Information Geometry of Turbo and Low-Density Parity-Check Codes

Shiro Ikeda
Institute of Statistical Mathematics
Mitao-ku, Tokyo, 106-8569 Japan

Toshiyuki Tanaka
Tokyo Metropolitan University
Hachioji-shi, Tokyo, 192-0397 Japan

Shun-ichi Amari
RIKEN Brain Science Institute
Wako-shi, Saitama, 351-0198 Japan

Abstract

Since the proposal of turbo codes in 1993, many studies have appeared on this simple and new type of codes which give a powerful and practical performance of error correction. Although experimental results strongly support the efficiency of turbo codes, further theoretical analysis is necessary, which is not straightforward. It is pointed out that turbo codes share essentially similar structure with low-density parity-check (LDPC) codes, with Pearl's belief propagation in a belief diagram, and with the Bethé approximation in statistical physics. Therefore, the mathematical structure, which lies behind turbo codes, will reveal the mystery of those similar iterative methods. In this paper, we recapture and extend the geometrical theory by Richardson in a more sophisticated framework of information geometry of dual affine connections, focusing on turbo and LDPC decoding. It gives a new approach to further analysis, and helps their intuitive understanding. We reveal some properties of these codes in the proposed framework, including the stability and error analysis. Based on the error analysis, we finally propose a correction term for improving the approximation.

Index Terms

turbo codes, low-density parity-check (LDPC) codes, information geometry, belief propagation

I. Introduction

The idea of turbo codes has been extensively studied since it was introduced in 1993[1]. The simple iterative decoding algorithm of turbo codes performs close to the optimum theoretical bound of error correction. However, the main properties so far obtained are mostly empirical, except for Richardson's geometrical framework[2]. The essence of turbo codes is not yet fully understood theoretically.

In addition to the experimental studies, clues to their essence have been sought in other methods. Since there are some iterative methods, which are closely related to turbo codes, theoretical results of those methods were expected to give further understanding of turbo codes. One of such methods is another class of error correcting codes called the low-density parity-check (LDPC) codes, which was originally proposed by Gallager[3] and rediscovered by MacKay[4]. Other methods related to turbo codes have been found in various fields, such as artificial intelligence and statistical physics. McEliece et al. showed that the turbo decoding algorithm is equivalent to the belief propagation algorithm[5], applied to a belief diagram with loops[6], [7]. MacKay demonstrated that the LDPC decoding algorithm is also equivalent to the belief propagation algorithm[4], while Kabashima and Saad pointed out that the iterative process of Bethé approximation in statistical physics is the same as that of the belief propagation algorithm [8], [9]. However, the efficacy of these methods is also a sort of mystery, and their findings do not help us clarify the mathematical structure of turbo codes.

Richardson[2] initiated a geometrical theory of turbo decoding. In this framework, he proved the existence of a fixed point by utilizing Brouwer's fixed point theorem, gave a condition for the fixed point to be unique, and analyzed when the fixed point is locally stable. This is good start but still many properties are not fully understood. Another theoretical approach is the density evolution[10], which describes the time evolution of message distribution. This is a powerful tool providing quantitative prediction of performance of codes, but it is not easy to have insights

for the mathematical structure of the codes. Therefore, further intuitive understanding of the iterative decoding algorithms is necessary. A hope will be found in information geometry[11], [12] which studies intrinsic geometrical structures existing in families of probability distributions. Along this line we extended the geometrical framework of Richardson to analyze iterative decoding algorithms, especially those for turbo and LDPC codes in a unified framework, and help their intuitive understanding. The framework is general so that main results are applicable to related iterative algorithms.

The ideal goal of iterative decoding algorithms is the maximization of the posterior marginals (MPM). However, since exact MPM decoding is usually computationally intractable, it is approximated using iterative algorithms. The algorithms are elucidated by using the e–projection and m–projection in information geometry together with the generalized Pythagorean theorem. Here, the Kullback-Leibler divergence, Fisher information, and the skewness tensor play fundamental roles. The equilibrium of the iterative algorithms is analyzed and its local stability condition is given in geometrical terms. These are regarded as a new formulation and elucidation of Richardson's framework.

We further analyze the accuracy of approximation or the decoding error in terms of e– and m–curvatures. The error term is given in an explicit form, so that the terms can be used to improve the decoding results. We give an explicit algorithm for the improvement. This is also used as a design principle of LDPC codes, and show why LDPC codes work so well. We finally touch upon the "free energy" in the statistical physics approach[9], [13].

The outline of the paper is as follows. In section II, we give the original schemes of turbo and LDPC codes. The basic strategy of MPM decoding is given in section III. Section IV introduces the information geometry. Sections V and VI describe the information geometry of turbo and LDPC decoding, respectively. Decoding errors are analyzed in section VII, and finally conclusion is given with some discussions for future perspectives in section IX.

II. Original Definitions of Turbo and LDPC Codes

A. Turbo Codes

1) Encoding: The idea of turbo codes is illustrated in Fig.1. Let $\boldsymbol{x} = (x_1, \cdots, x_N)^T, x_i \in \{-1, +1\}$ be the information bits to be transmitted. We assume a binary symmetric channel (BSC) with bit-error rate σ, and it is easy to generalize the results to any memoryless channel (see Appendix I). Turbo codes use two encoders, Encoders 1 and 2 in the figure, which generate two sets of parity bits in the encoding process. We denote them by $\boldsymbol{y}_1 = (y_{11}, \cdots, y_{1L})^T$ and $\boldsymbol{y}_2 = (y_{21}, \cdots, y_{2L})^T, y_{1j}, y_{2j} \in \{-1, +1\}$. Each set of parity bits $\boldsymbol{y}_r$, $r = 1, 2$, is a function of $\boldsymbol{x}$ and is represented as $\boldsymbol{y}_r(\boldsymbol{x})$ when an explicit expression is necessary. The set of these codes $(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2)$ are transmitted through the BSC, and a receiver observes their noisy version, $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2)$, $\tilde{x}_i, \tilde{y}_{1j}, \tilde{y}_{2j} \in \{-1, +1\}$.

2) Decoding: Turbo codes handle the case where direct decoding with $(\tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2)$ as a single set of parity bits is intractable, while soft decoding with each of $\tilde{\boldsymbol{y}}_1$, $\tilde{\boldsymbol{y}}_2$ is tractable. Two decoders are used for the decoding, Decoders 1 and 2 in the figure. Decoder 1 infers the original information bits, $\boldsymbol{x}$, from $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1)$, and Decoder 2 does the same from $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_2)$. The inferences of these two decoders may differ initially, and a better inference is searched for through iterative information exchange.

Let us define the following variables (see [2]) with the use of the conditional probabilities $p(\tilde{\boldsymbol{x}}|\boldsymbol{x})$ and $p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})$, $r = 1, 2$,

$$lx_i \stackrel{\text{def}}{=} \ln \frac{\sum_{\{\boldsymbol{x}:x_i=+1\}} p(\tilde{\boldsymbol{x}}|\boldsymbol{x})}{\sum_{\{\boldsymbol{x}:x_i=-1\}} p(\tilde{\boldsymbol{x}}|\boldsymbol{x})} = \ln \frac{p(\tilde{x}_i|x_i=+1)}{p(\tilde{x}_i|x_i=-1)},$$

$$ly_{rj} \stackrel{\text{def}}{=} \ln \frac{\sum_{\{\boldsymbol{x}:y_{rj}=+1\}} p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})}{\sum_{\{\boldsymbol{x}:y_{rj}=-1\}} p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})} = \ln \frac{p(\tilde{y}_{rj}|y_{rj}=+1)}{p(\tilde{y}_{rj}|y_{rj}=-1)},$$

$$L_r\boldsymbol{x} \stackrel{\text{def}}{=} F(l\boldsymbol{x}, l\boldsymbol{y}_r) = \left\{ \ln \frac{\sum_{\{\boldsymbol{x}:x_i=+1\}} p(\tilde{\boldsymbol{x}}|\boldsymbol{x}) p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})}{\sum_{\{\boldsymbol{x}:x_i=-1\}} p(\tilde{\boldsymbol{x}}|\boldsymbol{x}) p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})} \right\}. \tag{1}$$

The turbo decoding algorithm makes use of two slack variables, $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2 \in \mathcal{R}^N$, called "extrinsic variables," for exchanging information between the decoders. The algorithm is given as follows.

Turbo decoding (Original)

1) Let $\boldsymbol{\xi}_1 = 0$ and set $t = 1$.
2) Calculate $L_1\boldsymbol{x}^{(t)} = F((l\boldsymbol{x} + \boldsymbol{\xi}_1), l\boldsymbol{y}_1)$ from eq.(1) and update $\boldsymbol{\xi}_2$:
$$\boldsymbol{\xi}_2 = L_1\boldsymbol{x}^{(t)} - (l\boldsymbol{x} + \boldsymbol{\xi}_1).$$
3) Calculate $L_2\boldsymbol{x}^{(t)} = F((l\boldsymbol{x} + \boldsymbol{\xi}_2), l\boldsymbol{y}_2)$ from eq.(1) and update $\boldsymbol{\xi}_1$:
$$\boldsymbol{\xi}_1 = L_2\boldsymbol{x}^{(t)} - (l\boldsymbol{x} + \boldsymbol{\xi}_2).$$
4) Iterate 2 and 3 by incrementing t by one, until $L_1\boldsymbol{x}^{(t)} = L_2\boldsymbol{x}^{(t)} = L_1\boldsymbol{x}^{(t+1)} = L_2\boldsymbol{x}^{(t+1)}$.

Ideally, steps 2 and step 3 would be iterated until convergence is achieved, but in practice, the number of iterations is fixed at less than 20.

B. LDPC Codes

1) Encoding: Figure 2 illustrates the structure of LDPC codes. Let $\boldsymbol{s} = (s_1, \cdots, s_M)^T$, $s_i \in \{0, 1\}$, be the information bits. Although we use different notations than for turbo codes, it will soon become clear that the problems are formulated in the unified view, i.e., estimating $\boldsymbol{x}$ from an observed $\tilde{\boldsymbol{y}}$. To compose the generator and parity check matrices, two sparse matrices, $C_1 \in \{0, 1\}^{K \times M}$ and $C_2 \in \{0, 1\}^{K \times K}$ are prepared, where C_2 is invertible in the modulo 2 arithmetic. They are shared by the sender and the receiver. The parity check matrix is

$$H = (C_1 \; C_2), \quad H \in \{0, 1\}^{K \times N},$$

where $N = M + K$. The generator matrix, $G^T \in \{0, 1\}^{N \times M}$, is given by

$$G^T = \begin{pmatrix} E_M \\ C_2^{-1} C_1 \end{pmatrix} \mod 2,$$

where E_M is an identity matrix of size M. The codeword, $\boldsymbol{u} = (u_1, \cdots, u_N)^T$, is generated from $\boldsymbol{s}$:

$$\boldsymbol{u} = G^T \boldsymbol{s} \mod 2.$$

From the definition of G^T, the first M bits of $\boldsymbol{u}$ are identical to $\boldsymbol{s}$, and $\boldsymbol{u}$ is sent through the channel. We also assume a BSC with bit-error rate σ. Codeword $\boldsymbol{u}$ is disturbed and received as $\tilde{\boldsymbol{u}}$. Let the noise vector be $\boldsymbol{x} = (x_1, \cdots, x_N)^T$, $x_i \in \{0, 1\}$, and received bits $\tilde{\boldsymbol{u}}$ are

$$\tilde{\boldsymbol{u}} = \boldsymbol{u} + \boldsymbol{x} \mod 2.$$

LDPC decoding estimates $\boldsymbol{s}$ from $\tilde{\boldsymbol{u}}$, which is equivalent to estimating noise vector $\boldsymbol{x}$, since $\boldsymbol{s}$ is given by the first M bits of $\tilde{\boldsymbol{u}} + \boldsymbol{x}$ (mod 2). In the decoding process, parity check matrix $H = \{h_{ij}\} = (C_1\ C_2) \in \{0, 1\}^{K \times N}$ is used; it satisfies the equality $HG^T = O$. Syndrome vector $\boldsymbol{y} = (y_1, \cdots, y_K)^T$ is calculated using $\boldsymbol{y} = H\tilde{\boldsymbol{u}}$. When noise is $\boldsymbol{x}$, the syndrome $\boldsymbol{y}$ is

$$\boldsymbol{y}(\boldsymbol{x}) = H\tilde{\boldsymbol{u}} = H(\boldsymbol{u} + \boldsymbol{x}) = HG^T\boldsymbol{s} + H\boldsymbol{x} = H\boldsymbol{x} \mod 2.$$

When $\tilde{\boldsymbol{y}}$ is the observed syndrome, the decoding problem is to estimate $\boldsymbol{x}$ that satisfies $\tilde{\boldsymbol{y}} = \boldsymbol{y}(\boldsymbol{x})$.

2) Decoding: The iterative decoding algorithm for LDPC codes is described elsewhere [3], [4], [9]. It consists of two steps: the "horizontal step" and the "vertical step." We describe them using the following two sets of probability distributions,

$$\{q_{ri}^{(0)}, q_{ri}^{(1)}\}, \{p_{ri}^{(0)}, p_{ri}^{(1)}\}, \quad q_{ri}^{(0)} + q_{ri}^{(1)} = 1, \quad p_{ri}^{(0)} + p_{ri}^{(1)} = 1,$$

for pairs of indices (r, i), $r = 1, \cdots, K$, $i = 1, \cdots, N$, such that $h_{ri} = 1$.

LDPC decoding (Original)

Initialization: Set $p_{ri}^{(0)} = 1 - \sigma$ and $p_{ri}^{(1)} = \sigma$ for pairs of indices (r, i) such that $h_{ri} = 1$.

Horizontal step: Update $\{q_{ri}^{(0)}, q_{ri}^{(1)}\}$ as follows. Note that the terms indexed by pairs of (r, i) appear in the summations and products only if $h_{ri} = 1$.

$$lq_{ri} = \ln \frac{\sum_{\boldsymbol{x}: x_i = 1} \left\{ p(\tilde{y}_r | \boldsymbol{x}) \prod_{i': i' \neq i, h_{ri'} = 1} p_{ri'}^{(x_{i'})} \right\}}{\sum_{\boldsymbol{x}: x_i = 0} \left\{ p(\tilde{y}_r | \boldsymbol{x}) \prod_{i': i' \neq i, h_{ri'} = 1} p_{ri'}^{(x_{i'})} \right\}},$$

$$q_{ri}^{(0)} = \frac{1}{e^{lq_{ri}} + 1}, \quad q_{ri}^{(1)} = \frac{e^{lq_{ri}}}{e^{lq_{ri}} + 1}.$$

Vertical step: Update $\{p_{ri}^{(0)}, p_{ri}^{(1)}\}$ as follows.

$$lp_{ri} = \ln \frac{\sigma}{1 - \sigma} + \ln \frac{\prod_{r': r' \neq r, h_{r'i} = 1} q_{r'i}^{(1)}}{\prod_{r': r' \neq r, h_{r'i} = 1} q_{r'i}^{(0)}},$$

$$p_{ri}^{(0)} = \frac{1}{e^{lp_{ri}} + 1}, \quad p_{ri}^{(1)} = \frac{e^{lp_{ri}}}{e^{lp_{ri}} + 1}.$$

Convergence: Stop when

$$lp_i = \ln \frac{\sigma}{1-\sigma} + \ln \frac{\prod_{r':h_{r'i}=1} q^{(1)}_{r'i}}{\prod_{r':h_{r'i}=1} q^{(0)}_{r'i}}.$$

When the algorithm achieves convergence, the estimate of $\boldsymbol{x}$ is obtained as,

$$\hat{x}_i = \begin{cases} 1, & lp_i \geq 0, \\ 0, & lp_i < 0, \end{cases} \qquad i = 1, \cdots, N.$$

III. Formulation of MPM Decoding

A. Unified View of Turbo and LDPC Decoding

The ideal goal of either turbo and LDPC decoding is MPM decoding. We first define MPM decoding in a unified setting of turbo and LDPC decoding, and its specific form in each decoding is explained in the following subsections. For the rest of the paper, we use the bipolar, i.e., $\{-1, +1\}$, expression for each bit x_i, y_i, $\tilde{x}_i$, and $\tilde{y}_i$.

The decoding problem is generally solved based on the posterior distribution of $\boldsymbol{x}$ conditioned on the observed codeword or syndrome vector, i.e., $p(\boldsymbol{x}|\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2)$ in turbo codes and $p(\boldsymbol{x}|\tilde{\boldsymbol{y}})$ in LDPC codes. The posterior distribution of $\boldsymbol{x}$ is expressed as

$$q(\boldsymbol{x}) = C \exp(c_0(\boldsymbol{x}) + c_1(\boldsymbol{x}) + \cdots + c_K(\boldsymbol{x})), \tag{2}$$

where $c_0(\boldsymbol{x})$ consists of the linear terms of $\{x_i\}$; $c_r(\boldsymbol{x})$, $r = 1, \cdots, K$, contains higher order interactions of $\{x_i\}$, and the terms depend on the observed information, $\tilde{\boldsymbol{x}}$, $\tilde{\boldsymbol{y}}$. We assume $c_r(\boldsymbol{x}) \neq c_{r'}(\boldsymbol{x})$ for $r \neq r'$. Decoding is to estimate the information bits, $\boldsymbol{x}$, based on $q(\boldsymbol{x})$. One natural approach is MPM decoding. The MPM estimator minimizes the expected number of wrong bits in the decoded word. MPM decoding in the bipolar case is achieved by taking the expectation of $\boldsymbol{x}$ with respect to $q(\boldsymbol{x})$. Let $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_N)^T$ be the expectation of $\boldsymbol{x}$, and $\hat{\boldsymbol{x}}$ be the decoded MPM estimator. Then

$$\boldsymbol{\eta} = \sum_{\boldsymbol{x}} q(\boldsymbol{x})\boldsymbol{x}, \quad \hat{\boldsymbol{x}} = \mathrm{sgn}(\boldsymbol{\eta}), \tag{3}$$

where $\mathrm{sgn}(\cdot)$ works in a bitwise manner. The $\boldsymbol{\eta}$ gives the "soft decoding," and the sign of each soft bit η_i gives the final result, $\hat{x}_i$.

Let $q(x_i)$ be the marginal distribution of one component x_i in $q(\boldsymbol{x})$, and let $\mathit{\Pi}$ denote the operator of marginalization, which maps $q(\boldsymbol{x})$ to an independent distribution having the same marginal distributions:

$$\mathit{\Pi} \circ q(\boldsymbol{x}) \stackrel{\mathrm{def}}{=} \prod_{i=1}^{N} q(x_i).$$

The soft bit η_i depends only on the marginal distribution $q(x_i)$. Since $q(x_i)$ is a binary distribution, η_i has a one-to-one correspondence to $q(x_i)$. Therefore, soft decoding is equivalent to the marginalization of $q(\boldsymbol{x})$. The marginalization of $q(\boldsymbol{x})$ generally needs summation over all possible $\boldsymbol{x}$ but one x_i, and it is computationally not

tractable in the case of turbo and LDPC decoding, where the length of $\boldsymbol{x}$ is more than a few hundred. Instead of marginalizing the entire $q(\boldsymbol{x})$ in (2), we decompose it into simple submodels, $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)$, $r=1,\cdots,K$,

$$p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)=\exp(c_0(\boldsymbol{x})+\boldsymbol{\zeta}_r\cdot\boldsymbol{x}+c_r(\boldsymbol{x})-\varphi_r(\boldsymbol{\zeta}_r)), \tag{4}$$

where $\varphi_r(\boldsymbol{\zeta}_r)$ is the normalization factor. Each $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)$ includes only one nonlinear term $c_r(\boldsymbol{x})$, and the linear part $c_0(\boldsymbol{x})$ of $\boldsymbol{x}$ is adjusted further through $\boldsymbol{\zeta}_r$, which we intend to approximate the effect of the other $c_{r'}(\boldsymbol{x}), r'\neq r$. We thus have K component decoders, each of which decodes $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r), r=1,\cdots,K$. The parameter $\boldsymbol{\zeta}_r$ plays the role of a window through which information from the other decoders, $r'\neq r$, is exchanged. The idea is to adjust $\{\boldsymbol{\zeta}_r\}$ through iterative information exchange to approximate the overall $\mathit{\Pi}\circ q(\boldsymbol{x})$ with $\mathit{\Pi}\circ p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)$. We assume that the marginalization or soft decoding is tractable for any $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)$.

B. Turbo Decoding

In this subsection, the concrete forms of eqs.(2) and (4) for turbo codes are derived. In turbo codes, the receiver observes a noisy version of $(\boldsymbol{x},\boldsymbol{y}_1,\boldsymbol{y}_2)$ as $(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}_1,\tilde{\boldsymbol{y}}_2)$. We can easily derive the following relation from the assumption of a memoryless channel,

$$p(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}_1,\tilde{\boldsymbol{y}}_2|\boldsymbol{x})=p(\tilde{\boldsymbol{x}}|\boldsymbol{x})p(\tilde{\boldsymbol{y}}_1|\boldsymbol{x})p(\tilde{\boldsymbol{y}}_2|\boldsymbol{x}).$$

The Bayes posterior distribution $p(\boldsymbol{x}|\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}_1,\tilde{\boldsymbol{y}}_2)$ is defined with a prior distribution $\omega(\boldsymbol{x})$ of $\boldsymbol{x}$. In this paper, we consider the uniform prior, where $\omega(\boldsymbol{x})=1/2^N$, and the Bayes posterior distribution is derived as,

$$p(\boldsymbol{x}|\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}_1,\tilde{\boldsymbol{y}}_2)=\frac{p(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}_1,\tilde{\boldsymbol{y}}_2|\boldsymbol{x})\omega(\boldsymbol{x})}{\sum_{\boldsymbol{x}}p(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}_1,\tilde{\boldsymbol{y}}_2|\boldsymbol{x})\omega(\boldsymbol{x})}=\frac{p(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}_1,\tilde{\boldsymbol{y}}_2|\boldsymbol{x})}{\sum_{\boldsymbol{x}}p(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}_1,\tilde{\boldsymbol{y}}_2|\boldsymbol{x})}. \tag{5}$$

Since we consider BSC, where each bit is flipped independently with the same probability, $p(\tilde{\boldsymbol{x}}|\boldsymbol{x})$ and $p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})$ have the form of

$$p(\tilde{\boldsymbol{x}}|\boldsymbol{x})=\exp(\beta\tilde{\boldsymbol{x}}\cdot\boldsymbol{x}-N\psi(\beta)),\quad \psi(\beta)=\ln\left(e^{-\beta}+e^{\beta}\right)$$

$$p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})=\exp(\beta\tilde{\boldsymbol{y}}_r\cdot\boldsymbol{y}_r(\boldsymbol{x})-L\psi(\beta)),\quad r=1,2.$$

Positive real number β is called the inverse temperature in statistical physics and is related to σ by

$$\sigma=\frac{1}{2}(1-\tanh\beta),$$

$\beta\to 0$ as $\sigma\to 1/2$, and $\beta\to\infty$ as $\sigma\to 0$. Let us define

$$c_0(\boldsymbol{x})\stackrel{\text{def}}{=}\beta\tilde{\boldsymbol{x}}\cdot\boldsymbol{x},\quad c_r(\boldsymbol{x})\stackrel{\text{def}}{=}\beta\tilde{\boldsymbol{y}}_r\cdot\boldsymbol{y}_r(\boldsymbol{x}),\quad r=1,2,$$

where $c_0(\boldsymbol{x})$ is linear in $\boldsymbol{x}$, and $\tilde{\boldsymbol{y}}_r\cdot\boldsymbol{y}_r(\boldsymbol{x})$ are polynomials in $\boldsymbol{x}$, representing higher order correlational components of many x_i's. The Bayes posterior distribution eq.(5) is rewritten as

$$\begin{aligned}p(\boldsymbol{x}|\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}_1,\tilde{\boldsymbol{y}}_2)&=C\exp(c_0(\boldsymbol{x})+\beta\tilde{\boldsymbol{y}}_1\cdot\boldsymbol{y}_1(\boldsymbol{x})+\beta\tilde{\boldsymbol{y}}_2\cdot\boldsymbol{y}_2(\boldsymbol{x}))\\&=C\exp(c_0(\boldsymbol{x})+c_1(\boldsymbol{x})+c_2(\boldsymbol{x})),\\C&\stackrel{\text{def}}{=}\frac{1}{\sum_{\boldsymbol{x}}\exp(c_0(\boldsymbol{x})+c_1(\boldsymbol{x})+c_2(\boldsymbol{x}))}.\end{aligned}$$

This distribution corresponds to $q(\boldsymbol{x})$ in eq.(2), when $K = 2$.

In turbo decoding, each of the two constituent decoders marginalizes its own posterior distribution of $\boldsymbol{x}$ derived from $p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_r|\boldsymbol{x}) = p(\tilde{\boldsymbol{x}}|\boldsymbol{x})p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})$, where a prior distribution of the form

$$\omega(\boldsymbol{x}; \boldsymbol{\zeta}_r) = \exp(\boldsymbol{\zeta}_r \cdot \boldsymbol{x} - \psi(\boldsymbol{\zeta}_r)), \quad \boldsymbol{\zeta}_r \in \mathcal{R}^N, \quad \psi(\boldsymbol{\zeta}_r) = \sum_i \ln\bigl(e^{-\zeta_r^i} + e^{\zeta_r^i}\bigr)$$

is used for taking information from the other decoder in the form of $\boldsymbol{\zeta}_r$. This is an independent distribution in which the guess of the other decoder is used. The posterior distribution of decoder r is defined as

$$\begin{aligned} p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r) &\stackrel{\text{def}}{=} p(\boldsymbol{x}|\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_r; \boldsymbol{\zeta}_r) = \frac{p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_r|\boldsymbol{x})\omega(\boldsymbol{x}; \boldsymbol{\zeta}_r)}{\sum_{\boldsymbol{x}} p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_r|\boldsymbol{x})\omega(\boldsymbol{x}; \boldsymbol{\zeta}_r)} \\ &= \exp(c_0(\boldsymbol{x}) + c_r(\boldsymbol{x}) + \boldsymbol{\zeta}_r \cdot \boldsymbol{x} - \varphi_r(\boldsymbol{\zeta}_r)), \\ \varphi_r(\boldsymbol{\zeta}_r) &\stackrel{\text{def}}{=} \ln \sum_{\boldsymbol{x}} \exp(c_0(\boldsymbol{x}) + c_r(\boldsymbol{x}) + \boldsymbol{\zeta}_r \cdot \boldsymbol{x}), \quad r = 1, 2. \end{aligned}$$

It is clear that $\boldsymbol{\zeta}_r$ plays the role of the window of information exchange, and that the information is used as a prior. This distribution is of the form of eq.(4).

C. LDPC Decoding

We reformulate the LDPC decoding in this subsection. The vectors $\boldsymbol{s}$, $\boldsymbol{u}$, $\tilde{\boldsymbol{u}}$, $\tilde{\boldsymbol{y}}$, and $\boldsymbol{x}$ are treated in the bipolar form, while G^T and H are still in the binary, i.e., $\{0, 1\}$, form. Note that 0 in the binary form corresponds to $+1$ in the bipolar form, and vice versa. Each bit y_r of syndrome vector $\boldsymbol{y}(\boldsymbol{x})$ is written as a higher order correlational product of $\{x_i\}$ in the bipolar form, that is, as a monomial in $\boldsymbol{x}$:

$$y_r(\boldsymbol{x}) = \prod_{j \in \mathcal{L}_r} x_j, \quad \mathcal{L}_r \stackrel{\text{def}}{=} \{j \mid h_{jr} = 1\},$$

where h_{jr} are elements of the parity-check matrix H.

We now consider the probability distribution of $\tilde{\boldsymbol{y}}$ conditioned on $\boldsymbol{x}$:

$$\begin{aligned} p(\tilde{\boldsymbol{y}}|\boldsymbol{x}) &= \exp(\rho\tilde{\boldsymbol{y}} \cdot \boldsymbol{y}(\boldsymbol{x}) - K\psi(\rho)) \\ &= \exp(c_1(\boldsymbol{x}) + \cdots + c_K(\boldsymbol{x}) - K\psi(\rho)), \\ c_r(\boldsymbol{x}) &\stackrel{\text{def}}{=} \rho\tilde{y}_r y_r(\boldsymbol{x}). \end{aligned} \tag{6}$$

When ρ is large, the probability of $\tilde{\boldsymbol{y}}$ is concentrated in the subset satisfying $\tilde{\boldsymbol{y}} = \boldsymbol{y}(\boldsymbol{x})$ because $\tilde{\boldsymbol{y}} \cdot \boldsymbol{y}(\boldsymbol{x})$ is maximized in the set. "Hard inference" of LDPC codes is to search for the $\boldsymbol{x}$ that exactly satisfies the parity check equations:

$$\tilde{\boldsymbol{y}} = \boldsymbol{y}(\boldsymbol{x}).$$

This is the maximum likelihood decoding. However, the procedure is intractable for large K, and we use "soft inference" which searches for the $\boldsymbol{x}$ that maximizes probability distribution $p(\tilde{\boldsymbol{y}}|\boldsymbol{x})$ in eq.(6) with a moderate ρ when syndrome $\tilde{\boldsymbol{y}}$ is observed. This is convenient for MPM decoding, which minimizes the bitwise error rate. Although "soft inference" approaches "hard inference" as ρ becomes larger, probability distribution $p(\tilde{\boldsymbol{y}}|\boldsymbol{x})$ is not smooth for a large ρ, and iteration is difficult. One approach is to begin with a moderate ρ, say $\rho = 1$ or 2, and to

increase it (annealing). Empirical studies has shown that the "soft inference" with a fixed ρ has sufficiently good performance[4].

Note that noise $\boldsymbol{x}$ is bitwise independent, and its error rate is given by $\sigma = (1/2)(1 - \tanh\beta)$. Consequently, we have the prior distribution $\omega(\boldsymbol{x})$:

$$\omega(\boldsymbol{x}) = \exp(\beta \mathbf{1}_N \cdot \boldsymbol{x} - N\psi(\beta)) = \exp(c_0(\boldsymbol{x}) - N\psi(\beta))$$
$$c_0(\boldsymbol{x}) \stackrel{\text{def}}{=} \beta \sum_{i=1}^{N} x_i, \quad \mathbf{1}_N = (\underbrace{1, \cdots, 1}_{N})^T.$$

As a result, the Bayes posterior distribution becomes

$$p(\boldsymbol{x}|\tilde{\boldsymbol{y}}) = \frac{p(\tilde{\boldsymbol{y}}|\boldsymbol{x})\omega(\boldsymbol{x})}{\sum_{\boldsymbol{x}} p(\tilde{\boldsymbol{y}}|\boldsymbol{x})\omega(\boldsymbol{x})} = C \exp(c_0(\boldsymbol{x}) + c_1(\boldsymbol{x}) + \cdots + c_K(\boldsymbol{x})).$$

This is equivalent to $q(\boldsymbol{x})$ in eq.(2).

In the horizontal and vertical steps of LDPC decoding, marginalization is carried out based on distribution $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)$, which is calculated from $p(\tilde{y}_r|\boldsymbol{x})$ and prior $\omega(\boldsymbol{x}; \boldsymbol{\zeta}_r)$. The parameter specifying the prior $\boldsymbol{\zeta}_r$ is obtained through the window for taking information from the other decoders' r's. We have

$$p(\tilde{y}_r|\boldsymbol{x}) = \exp(c_r(\boldsymbol{x}) - \psi(\beta))$$
$$\omega(\boldsymbol{x}; \boldsymbol{\zeta}_r) = \exp((\beta \mathbf{1}_N + \boldsymbol{\zeta}_r) \cdot \boldsymbol{x} - \psi(\beta \mathbf{1}_N + \boldsymbol{\zeta}_r)), \quad \boldsymbol{\zeta}_r \in \mathcal{R}^N,$$
$$p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r) = p(\boldsymbol{x}|\tilde{y}_r; \boldsymbol{\zeta}_r) = \frac{p(\tilde{y}_r|\boldsymbol{x})\omega(\boldsymbol{x}; \boldsymbol{\zeta}_r)}{\sum_{\boldsymbol{x}} p(\tilde{y}_r|\boldsymbol{x})\omega(\boldsymbol{x}; \boldsymbol{\zeta}_r)}$$
$$= \exp(c_0(\boldsymbol{x}) + c_r(\boldsymbol{x}) + \boldsymbol{\zeta}_r \cdot \boldsymbol{x} - \varphi_r(\boldsymbol{\zeta}_r))$$
$$\varphi_r(\boldsymbol{\zeta}_r) \stackrel{\text{def}}{=} \ln \sum_{\boldsymbol{x}} \exp(c_0(\boldsymbol{x}) + c_r(\boldsymbol{x}) + \boldsymbol{\zeta}_r \cdot \boldsymbol{x}), \quad r = 1, 2, \cdots, K.$$

This coincides with the formulation in eq.(4). The above argument shows that the LDPC decoding falls into the general framework given in section III-A.

IV. Information Geometry of Probability Distributions

The preliminaries from information geometry[11], [12] are given in this section.

A. Manifolds of Probability Distributions and e–flat, m–flat Submanifolds

Consider the family of all the probability distributions over $\boldsymbol{x}$. We denote it by S:

$$S = \left\{ p(\boldsymbol{x}) \;\middle|\; p(\boldsymbol{x}) > 0, \boldsymbol{x} \in \{-1, +1\}^N, \sum_{\boldsymbol{x}} p(\boldsymbol{x}) = 1 \right\}.$$

This is the set of all distributions over 2^N atoms $\boldsymbol{x}$. The family S has $(2^N - 1)$ degrees of freedom and is a $(2^N - 1)$–dimensional manifold belonging to the exponential family[12], [14].

In order to prove this, we introduce random variables

$$\delta_{i_1 \cdots i_N}(\boldsymbol{x}) = \begin{cases} 1, & \text{when } \boldsymbol{x} = (i_1, \cdots, i_N)^T, \text{ where } i_k \in \{-1, +1\}, \ k = 1, \cdots, N \\ 0, & \text{otherwise,} \end{cases}$$

Any $p(\boldsymbol{x}) \in S$ is expanded in the following form:

$$p(\boldsymbol{x}) = \sum p_{i_1 \cdots i_N} \delta_{i_1 \cdots i_N}(\boldsymbol{x}), \tag{7}$$

where $p_{i_1 \cdots i_N} = \Pr\{x_1 = i_1, \cdots, x_N = i_N\}$, which shows $p(\boldsymbol{x}) \in S$ is parameterized by 2^N variables $\{p_{i_1 \cdots i_N}\}$. Since $\sum p(\boldsymbol{x}) = 1$, the family S has $(2^N - 1)$ degrees of freedom.

Similarly, $\ln p(\boldsymbol{x})$ is expanded:

$$\ln p(\boldsymbol{x}) = \sum_{i_1 \cdots i_N} (\ln p_{i_1 \cdots i_N}) \delta_{i_1 \cdots i_N}(\boldsymbol{x}).$$

Since the degrees of freedom are $(2^N - 1)$, we set $\boldsymbol{\theta} = \{\theta_{i_1 \cdots i_N} \mid (i_1 \cdots i_N) \neq (-1 \cdots -1)\}$,

$$\theta_{i_1 \cdots i_N} = \ln \frac{p_{i_1 \cdots i_N}}{p_{-1 \cdots -1}}.$$

and rewrite eq.(7) as

$$p(\boldsymbol{x}; \boldsymbol{\theta}) = \exp\Big(\sum_{i_1 \cdots i_N} \theta_{i_1 \cdots i_N} \delta_{i_1 \cdots i_N}(\boldsymbol{x}) - \varphi(\boldsymbol{\theta}) \Big),$$

where

$$\varphi(\boldsymbol{\theta}) = -\ln \Pr\{x_1 = \cdots = x_N = -1\}.$$

This shows S is an exponential family whose natural, or canonical, coordinate system is $\boldsymbol{\theta}$.

The expectations of random variables $\delta_{i_1 \cdots i_N}(\boldsymbol{x})$ are

$$\eta_{i_1 \cdots i_N} = E_p[\delta_{i_1 \cdots i_N}(\boldsymbol{x})] = p_{i_1 \cdots i_N}.$$

They form another coordinate system of S that specifies $p(\boldsymbol{x})$,

$$\boldsymbol{\eta} = \{\eta_{i_1 \cdots i_N}\}, \ (i_1 \cdots i_N) \neq (-1 \cdots -1).$$

Since S is an exponential family, it naturally has two affine structures: the exponential– or e–affine structure and the mixture– or m–affine structure. Equivalent structures were also used by Richardson[2]. When manifold S is regarded as an affine space in $\ln p(\boldsymbol{x})$, it is e–affine, and $\boldsymbol{\theta}$ gives the e–affine coordinate system. Similarly, when manifold S is regarded as an affine space in $p(\boldsymbol{x})$, it is m–affine, and $\boldsymbol{\eta}$ gives the m–affine coordinate system. They are dually coupled with respect to the Riemannian structure given by the Fisher information matrix, which will be introduced below.

First we define the e–flat and m–flat submanifolds of S.

e–flat submanifold: Submanifold $M \subset S$ is said to be e–flat, when the following $r(\boldsymbol{x}; t)$ belongs to M for all $t \in [0, 1], q(\boldsymbol{x}), p(\boldsymbol{x}) \in M$.

$$\ln r(\boldsymbol{x}; t) = (1 - t) \ln q(\boldsymbol{x}) + t \ \ln p(\boldsymbol{x}) + c(t), \qquad t \in R,$$

where $c(t)$ is the normalization factor. Obviously, $\{r(\boldsymbol{x}; t) | t \in [0, 1]\}$ is an exponential family connecting two distributions, $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$. In particular, when an e–flat submanifold is a one–dimensional curve,

it is called an e–geodesic. The above $\{r(\boldsymbol{x};t)|t\in[0,1]\}$ is the e–geodesic connecting $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$. In terms of the e–affine coordinates, $\boldsymbol{\theta}$, a submanifold M is e–flat when it is linear in $\boldsymbol{\theta}$.

m–flat submanifold: Submanifold $M\subset S$ is said to be m–flat when the following mixture $r(\boldsymbol{x};t)$ belongs to M for all $t\in[0,1], q(\boldsymbol{x}), p(\boldsymbol{x})\in M$.

$$r(\boldsymbol{x};t)=(1-t)q(\boldsymbol{x})+tp(\boldsymbol{x}),\qquad t\in[0,1].$$

When an m–flat submanifold is a one–dimensional curve, it is called an m–geodesic. Hence, the above mixture family is the m–geodesic connecting them. In terms of the m–affine coordinates, $\boldsymbol{\eta}$, a submanifold M is m–flat when it is linear in $\boldsymbol{\eta}$.

B. KL–divergence, Fisher Metric, and Generalized Pythagorean Theorem

Manifold S has a Riemannian metric given by the Fisher information matrix I. We begin with the Kullback-Leibler divergence, $D[\cdot;\cdot]$, defined by

$$D[q(\boldsymbol{x});p(\boldsymbol{x})]=\sum_{\boldsymbol{x}}q(\boldsymbol{x})\ln\frac{q(\boldsymbol{x})}{p(\boldsymbol{x})}.$$

The KL–divergence satisfies $D[q(\boldsymbol{x});p(\boldsymbol{x})]\geq 0$, and $D[q(\boldsymbol{x});p(\boldsymbol{x})]=0$ when and only when $q(\boldsymbol{x})=p(\boldsymbol{x})$ holds for every $\boldsymbol{x}$. Although symmetry $D[q;p]=D[p;q]$ does not hold generally, it is regarded as an asymmetric squared distance.

Consider two nearby distributions $p(\boldsymbol{x};\boldsymbol{\theta})$ and $p(\boldsymbol{x};\boldsymbol{\theta}+d\boldsymbol{\theta})$, specified by coordinates $\boldsymbol{\theta}$ and $\boldsymbol{\theta}+d\boldsymbol{\theta}$ in any coordinate system. From Taylor expansion, their KL–divergence is given by the quadratic form

$$D[p(\boldsymbol{x};\boldsymbol{\theta});p(\boldsymbol{x};\boldsymbol{\theta}+d\boldsymbol{\theta})]=\frac{1}{2}d\boldsymbol{\theta}^T I(\boldsymbol{\theta})d\boldsymbol{\theta},$$

where $I(\boldsymbol{\theta})$ is the Fisher information matrix defined by

$$I(\boldsymbol{\theta})=\sum_{\boldsymbol{x}}p(\boldsymbol{x};\boldsymbol{\theta})\partial_{\boldsymbol{\theta}}\ln p(\boldsymbol{x};\boldsymbol{\theta})(\partial_{\boldsymbol{\theta}}\ln p(\boldsymbol{x};\boldsymbol{\theta}))^T=-\sum_{\boldsymbol{x}}p(\boldsymbol{x};\boldsymbol{\theta})\partial_{\boldsymbol{\theta}\boldsymbol{\theta}}\ln p(\boldsymbol{x};\boldsymbol{\theta}),$$

where $\partial_{\boldsymbol{\theta}}$ represents the gradient operator (differentiation with respect to the components of $\boldsymbol{\theta}$). When the squared distance of a small line element $d\boldsymbol{\theta}$ starting from $\boldsymbol{\theta}$ is given by the quadratic form

$$ds^2=d\boldsymbol{\theta}^T G(\boldsymbol{\theta})d\boldsymbol{\theta},$$

the space is called a Riemannian manifold with the Riemannian metric tensor $G(\boldsymbol{\theta})$, which is a positive-definite matrix depending on $\boldsymbol{\theta}$. In the present case, the Fisher information matrix $I(\boldsymbol{\theta})$ plays the role of the Riemannian metric $G(\boldsymbol{\theta})$. Hence, the infinitesimal KL-divergence is regarded as half the square of the Riemannian distance.

The Riemannian metric also defines the orthogonality of two intersecting curves. Let $p(\boldsymbol{x};\boldsymbol{\theta}_1(t))$ and $p(\boldsymbol{x};\boldsymbol{\theta}_2(t))$ be two curves intersecting at $t=0$; that is, $\boldsymbol{\theta}_1(0)=\boldsymbol{\theta}_2(0)$. The tangent vectors of the curves at $t=0$ are represented by $\dot{\boldsymbol{\theta}}_1(t)$ and $\dot{\boldsymbol{\theta}}_2(t)$ by using the coordinates, where $\dot{\boldsymbol{\theta}}_i(t)=d\boldsymbol{\theta}_i(t)/dt$. The two curves are said to be orthogonal at their intersection $t=0$, when their inner product with respect to the Riemannian metric vanishes:

$$\langle\dot{\boldsymbol{\theta}}_1(0),\dot{\boldsymbol{\theta}}_2(0)\rangle=\dot{\boldsymbol{\theta}}_1(0)^T I(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}_2(0)=0.$$

Now we state the generalized Pythagoras theorem and projection theorem, which holds in a general dually flat manifold[12], and show the dual nature of the e– and m–structures with the Riemannian metric.

Theorem 1: Let $p(\boldsymbol{x})$, $q(\boldsymbol{x})$, and $r(\boldsymbol{x})$ be three distributions in S. When the m–geodesic connecting $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ is orthogonal at $q(\boldsymbol{x})$ to the e–geodesic connecting $q(\boldsymbol{x})$ and $r(\boldsymbol{x})$, the following relation holds

$$D[p(\boldsymbol{x}); r(\boldsymbol{x})] = D[p(\boldsymbol{x}); q(\boldsymbol{x})] + D[q(\boldsymbol{x}); r(\boldsymbol{x})].$$

Next we define m–projection.

Definition 1: Let M be an e–flat submanifold in S, and let $q(\boldsymbol{x}) \in S$. The point in M that minimizes the KL–divergence from $q(\boldsymbol{x})$ to M is denoted by

$$\Pi_M \circ q(\boldsymbol{x}) = \underset{p(\boldsymbol{x}) \in M}{\operatorname{argmin}} D[q(\boldsymbol{x}); p(\boldsymbol{x})]$$

and is called the m–projection of $q(\boldsymbol{x})$ to M.

Finally, the m–projection theorem follows.

Theorem 2: Let M be an e–flat submanifold in S, and let $q(\boldsymbol{x}) \in S$. The m–projection of $q(\boldsymbol{x})$ to M is unique and given by a point in M such that the m–geodesic connecting $q(\boldsymbol{x})$ and $\Pi_M \circ q$ is orthogonal to M at this point.

C. Legendre Transformation and Local Structure

Let $\boldsymbol{\theta}$ be the e–affine coordinate system of S. Every exponential family has the form

$$p(\boldsymbol{x}; \boldsymbol{\theta}) = \exp(c(\boldsymbol{x}) + \boldsymbol{\theta} \cdot \boldsymbol{x} - \varphi(\boldsymbol{\theta})).$$

The function $\varphi(\boldsymbol{\theta})$ is a convex function which is called the cumulant generating function in statistics, and the free energy function in statistical physics. The m–affine coordinate system $\boldsymbol{\eta}$ is given by its gradient.

$$\boldsymbol{\eta} = \partial_{\boldsymbol{\theta}} \varphi(\boldsymbol{\theta}),$$

where $\partial_{\boldsymbol{\theta}} = \partial / \partial \boldsymbol{\theta}$ is the gradient. This is the Legendre transformation; the dual potential, $\phi(\boldsymbol{\eta})$ is given

$$\varphi(\boldsymbol{\theta}) + \phi(\boldsymbol{\eta}) - \boldsymbol{\theta} \cdot \boldsymbol{\eta} = 0$$

and is the negative of the Shannon entropy,

$$\phi(\boldsymbol{\eta}) = \sum_{\boldsymbol{x}} p(\boldsymbol{x}; \boldsymbol{\eta}) \ln p(\boldsymbol{x}; \boldsymbol{\eta}).$$

The Fisher information matrix is given by the second derivative of φ,

$$I(\boldsymbol{\theta}) = \partial_{\boldsymbol{\theta}\boldsymbol{\theta}} \varphi(\boldsymbol{\theta}),$$

which is positive-definite. We have shown that the square of the local distance is given by

$$D[p(\boldsymbol{x}; \boldsymbol{\theta}); p(\boldsymbol{x}; \boldsymbol{\theta} + d\boldsymbol{\theta})] = D[p(\boldsymbol{x}; \boldsymbol{\theta} + d\boldsymbol{\theta}); d\boldsymbol{\theta}] = \frac{1}{2} d\boldsymbol{\theta}^T I(\boldsymbol{\theta}) d\boldsymbol{\theta}.$$

The third derivative of potential φ,

$$T = \partial_{\boldsymbol{\theta}\boldsymbol{\theta}\boldsymbol{\theta}} \varphi(\boldsymbol{\theta}),$$

is called the skewness tensor. It is a symmetric tensor of order three, and its components are calculated as

$$T_{ijk} = E_p[(x_i - \eta_i)(x_j - \eta_j)(x_k - \eta_k)],$$

where $E_p[\cdot]$ denotes expectation with respect to $p(\boldsymbol{x})$. The KL–divergence is expanded as

$$D[p(\boldsymbol{x};\boldsymbol{\theta}); p(\boldsymbol{x};\boldsymbol{\theta}+d\boldsymbol{\theta})] = \frac{1}{2} d\boldsymbol{\theta}^T I(\boldsymbol{\theta}) d\boldsymbol{\theta} + \frac{1}{6}(d\boldsymbol{\theta})^3 \circ T(\boldsymbol{\theta}),$$

where

$$(d\boldsymbol{\theta})^3 \circ T(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \sum_{i,j,k} d\theta_i d\theta_j d\theta_k T_{ijk}(\boldsymbol{\theta})$$

in the component form. This shows the local asymmetry of the KL–divergence:

$$D[p(\boldsymbol{x};\boldsymbol{\theta}); p(\boldsymbol{x};\boldsymbol{\theta}+d\boldsymbol{\theta})] - D[p(\boldsymbol{x};\boldsymbol{\theta}+d\boldsymbol{\theta}); p(\boldsymbol{x};\boldsymbol{\theta})] = \frac{1}{3}(d\boldsymbol{\theta})^3 \circ T(\boldsymbol{\theta}).$$

The skewness tensor plays a fundamental role in the analysis of decoding error.

D. Important Submanifolds and Marginalization

Now, we consider a submanifold, M_D, in which every joint distribution is decomposed as

$$p(\boldsymbol{x}) = \prod_{i=1}^{N} p(x_i), \quad p(\boldsymbol{x}) \in M_D.$$

All the bits of $\boldsymbol{x}$ are independent for a distribution in M_D. Since each bit takes one of $\{-1, +1\}$, $p(x_i)$ is a binomial distribution, and $p(\boldsymbol{x})$ belongs to an exponential family of the form

$$\begin{aligned} p(\boldsymbol{x};\boldsymbol{\theta}) &= \prod_{i=1}^{N} p(x_i;\theta_i) = \prod_{i=1}^{N} \exp(\theta_i x_i - \varphi(\theta_i)) = \exp(\boldsymbol{\theta}\cdot\boldsymbol{x} - \varphi(\boldsymbol{\theta})) \\ \varphi(\boldsymbol{\theta}) &= \sum_{i=1}^{N} \varphi(\theta_i) = \ln \sum_{i=1}^{N} \left(e^{-\theta_i} + e^{\theta_i}\right), \quad \boldsymbol{\theta} \in \mathcal{R}^N. \end{aligned} \tag{8}$$

The submanifold M_D is N-dimensional, with its e–affine coordinate system $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_N)^T$, which are the natural or canonical parameters in M_D. The other parameter (m–affine coordinate system) is the expectation parameter $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_N)^T$ defined by

$$\boldsymbol{\eta} = E_p[\boldsymbol{x}] = \sum_{\boldsymbol{x}} p(\boldsymbol{x};\boldsymbol{\theta})\boldsymbol{x}.$$

This is equivalent to the soft decoding in eq.(3). There is a simple one-to-one correspondence between $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$:

$$\partial_{\boldsymbol{\theta}}\varphi(\boldsymbol{\theta}) = \boldsymbol{\eta}, \quad \eta_i = \tanh(\theta_i), \quad \theta_i = \frac{1}{2}\ln\frac{1+\eta_i}{1-\eta_i}, \quad i = 1, \cdots, N.$$

Proposition 1: M_D is an e–flat submanifold of S.

Proof: M_D is a submanifold of S. Let $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathcal{R}^N$ and $p(\boldsymbol{x};\boldsymbol{\theta}), p(\boldsymbol{x};\boldsymbol{\theta}') \in M_D$. For any $\boldsymbol{\theta}, \boldsymbol{\theta}'$,

$$\begin{aligned} \ln r(\boldsymbol{x};t) &= (1-t)\ln p(\boldsymbol{x};\boldsymbol{\theta}) + t \ln p(\boldsymbol{x};\boldsymbol{\theta}') + c(\boldsymbol{\theta},\boldsymbol{\theta}';t) \\ &= ((1-t)\boldsymbol{\theta} + t\boldsymbol{\theta}')\cdot\boldsymbol{x} + c(\boldsymbol{\theta},\boldsymbol{\theta}';t). \end{aligned}$$

Let $\boldsymbol{u}(t) \stackrel{\text{def}}{=} (1-t)\boldsymbol{\theta} + t\boldsymbol{\theta}'$, and $r(\boldsymbol{x};t) = \exp(\boldsymbol{u}(t)\cdot\boldsymbol{x} - \varphi(\boldsymbol{u}(t)))$ belongs to M_D. ■

We now define a number of e–flat submanifolds that play important roles in the decoding algorithms. The first is the submanifold of $p_0(\boldsymbol{x};\boldsymbol{\theta})$ defined by

$$M_0 = \Big\{ p_0(\boldsymbol{x};\boldsymbol{\theta}) = \exp(c_0(\boldsymbol{x}) + \boldsymbol{\theta}\cdot\boldsymbol{x} - \varphi_0(\boldsymbol{\theta})) \;\Big|\; \boldsymbol{x}\in\{-1,+1\}^N, \boldsymbol{\theta}\in\mathcal{R}^N \Big\}.$$

Since $c_0(\boldsymbol{x})$ is linear in $\{x_i\}$, M_0 is identical to M_D. Let $c_0(\boldsymbol{x}) = \boldsymbol{\alpha}\cdot\boldsymbol{x}$, where $\boldsymbol{\alpha} = \beta\tilde{\boldsymbol{x}}$ for turbo codes and $\boldsymbol{\alpha} = \beta\mathbf{1}_N$ for LDPC codes. The new coordinate, $\boldsymbol{\theta}$ is obtained by shifting the old one, $\boldsymbol{\theta}_{\text{old}}$, in eq.(8) by $\boldsymbol{\alpha}$:

$$\boldsymbol{\theta} = \boldsymbol{\theta}_{\text{old}} - \boldsymbol{\alpha}, \quad \varphi_0(\boldsymbol{\theta}) = \varphi(\boldsymbol{\theta}+\boldsymbol{\alpha}).$$

We use the coordinates $\boldsymbol{\theta}$ as a coordinate system of M_0, in which information from the constituent decoders is integrated. We define the expectation parameter as $\boldsymbol{\eta}_0(\boldsymbol{\theta})$, which is another coordinate system of M_0 and is dual to $\boldsymbol{\theta}$:

$$\boldsymbol{\eta}_0(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \sum_{\boldsymbol{x}} p_0(\boldsymbol{x};\boldsymbol{\theta})\boldsymbol{x} = \partial_{\boldsymbol{\theta}}\varphi_0(\boldsymbol{\theta}). \tag{9}$$

Next, we consider the submanifold primarily responsible for only one $c_r(\boldsymbol{x})$. The submanifold, M_r, $r = 1,\cdots,K$ ($K = 2$ for turbo codes), is defined by

$$M_r = \Big\{ p_r(\boldsymbol{x};\boldsymbol{\zeta}_r) = \exp(c_0(\boldsymbol{x}) + c_r(\boldsymbol{x}) + \boldsymbol{\zeta}_r\cdot\boldsymbol{x} - \varphi_r(\boldsymbol{\zeta}_r)) \;\Big|\; \boldsymbol{x}\in\{-1,+1\}^N, \boldsymbol{\zeta}_r\in\mathcal{R}^N \Big\}.$$

Here, $\boldsymbol{\zeta}_r$ is the e–affine coordinate system or the natural parameters of M_r, through which information of the other decoders is integrated. M_r is also an e–flat submanifold of S. However, $M_r \neq M_0$ and $M_r \neq M_{r'}$, $r \neq r'$, because $c_r(\boldsymbol{x})$ includes higher order correlations of $\{x_i\}$ and $c_r(\boldsymbol{x}) \neq c_{r'}(\boldsymbol{x})$. The expectation parameter for M_r is defined as

$$\boldsymbol{\eta}_r(\boldsymbol{\zeta}_r) \stackrel{\text{def}}{=} \sum_{\boldsymbol{x}} p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)\boldsymbol{x} = \partial_{\boldsymbol{\zeta}_r}\varphi_r(\boldsymbol{\zeta}_r). \tag{10}$$

We show that the soft decoding is the m–projection to M_0 of the posterior distribution. Let us consider the m–projection of $q(\boldsymbol{x})$ to M_0. The derivative of $D[q(\boldsymbol{x}); p_0(\boldsymbol{x};\boldsymbol{\theta})]$ with respect to $\boldsymbol{\theta}$ is

$$\partial_{\boldsymbol{\theta}} D[q(\boldsymbol{x}); p_0(\boldsymbol{x};\boldsymbol{\theta})] = \partial_{\boldsymbol{\theta}}\varphi_0(\boldsymbol{\theta}) - \sum_{\boldsymbol{x}} q(\boldsymbol{x})\boldsymbol{x} = \boldsymbol{\eta}_0(\boldsymbol{\theta}) - \sum_{\boldsymbol{x}} q(\boldsymbol{x})\boldsymbol{x}.$$

By the definition of the m–projection, this vanishes at the projected point. Hence, the m–affine coordinate of the projected point $\boldsymbol{\theta}^*$ is given by $\boldsymbol{\eta}_0(\boldsymbol{\theta}^*) = \sum_{\boldsymbol{x}} q(\boldsymbol{x})\boldsymbol{x}$,

$$\eta_{0,i}(\theta_i^*) = \sum_{\boldsymbol{x}} q(\boldsymbol{x})x_i = \sum_{x_i} q(x_i)x_i,$$

which shows that the m–projection of $q(\boldsymbol{x})$ does not change the expectation of $\boldsymbol{x}$. This is equivalent to the soft decoding defined in eq.(3) (Fig.3).

V. Information Geometry of Turbo Decoding

The goal of turbo decoding is to obtain a good approximation of MPM decoding for $q(\boldsymbol{x}) = p(\boldsymbol{x}|\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2)$. Although the m–projection of $q(\boldsymbol{x})$ to M_0 is not tractable, the m–projection of any distribution $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r) \in M_r$, $r = 1, 2$ to M_0 is tractable with BCJR algorithm. Since each $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)$, $r = 1, 2$, is derived from $p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_r|\boldsymbol{x})$ and a prior $\omega(\boldsymbol{x}; \boldsymbol{\zeta}_r) \in M_D$, we can describe turbo decoding as a method to approximate the m–projection of $q(\boldsymbol{x})$ to M_0 by changing the prior of $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)$ iteratively and projecting $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)$ to M_0.

A. Information Geometrical Definition of Turbo Decoding

The turbo decoding algorithm in subsection II-A is rewritten in the information geometrical framework. It is convenient to use an adequate e–affine coordinate system of M for the m–projection of $q(\boldsymbol{x})$ to M. Let $\pi_M \circ q(\boldsymbol{x})$ denote the coordinate $\boldsymbol{\theta}$ of M corresponding to the m–projected distribution:

$$\pi_M \circ q(\boldsymbol{x}) = \underset{\boldsymbol{\theta} \in \mathcal{R}^N}{\operatorname{argmin}} D[q(\boldsymbol{x}); p(\boldsymbol{x}; \boldsymbol{\theta})].$$

Turbo decoding (information geometrical view)

1) Let $\boldsymbol{\zeta}_2^t = 0$ for $t = 0$. For $t = 0, 1, 2, \cdots$, compose $p_2(\boldsymbol{x}; \boldsymbol{\zeta}_2^t) \in M_2$ with prior $\boldsymbol{\zeta}_2^t$.
2) Perform m–projection of $p_2(\boldsymbol{x}; \boldsymbol{\zeta}_2^t)$ to M_0 as $\pi_{M_0} \circ p_2(\boldsymbol{x}; \boldsymbol{\zeta}_2^t)$, and update $\boldsymbol{\zeta}_1^{t+1}$ using

$$\boldsymbol{\zeta}_1^{t+1} = \pi_{M_0} \circ p_2(\boldsymbol{x}; \boldsymbol{\zeta}_2^t) - \boldsymbol{\zeta}_2^t. \tag{11}$$

3) Compose $p_1(\boldsymbol{x}; \boldsymbol{\zeta}_1^{t+1}) \in M_1$. Perform m–projection of $p_1(\boldsymbol{x}; \boldsymbol{\zeta}_1^{t+1})$ to M_0 as $\pi_{M_0} \circ p_1(\boldsymbol{x}; \boldsymbol{\zeta}_1^{t+1})$ and update $\boldsymbol{\zeta}_2^{t+1}$ using

$$\boldsymbol{\zeta}_2^{t+1} = \pi_{M_0} \circ p_1(\boldsymbol{x}; \boldsymbol{\zeta}_1^{t+1}) - \boldsymbol{\zeta}_1^{t+1}. \tag{12}$$

4) If $\pi_{M_0} \circ p_1(\boldsymbol{x}; \boldsymbol{\zeta}_1^{t+1}) \neq \pi_{M_0} \circ p_2(\boldsymbol{x}; \boldsymbol{\zeta}_2^{t+1})$, go to step 1.

To clarify this procedure, we introduce three auxiliary parameters $\boldsymbol{\theta}$, $\boldsymbol{\xi}_1$, and $\boldsymbol{\xi}_2$:

$$\boldsymbol{\theta} \stackrel{\text{def}}{=} \boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2, \quad \boldsymbol{\xi}_1 \stackrel{\text{def}}{=} \boldsymbol{\theta} - \boldsymbol{\zeta}_1 = \boldsymbol{\zeta}_2, \quad \boldsymbol{\xi}_2 \stackrel{\text{def}}{=} \boldsymbol{\theta} - \boldsymbol{\zeta}_2 = \boldsymbol{\zeta}_1.$$

The intuition behind this framework is as follows. Each of the higher order correlation term, $c_1(\boldsymbol{x})$ or $c_2(\boldsymbol{x})$ is included only in Decoder 1 or Decoder 2, respectively. Decoders 1 and 2 calculate, using the m–projection, the linear approximations $\boldsymbol{\xi}_1 \cdot \boldsymbol{x}$ and $\boldsymbol{\xi}_2 \cdot \boldsymbol{x}$ of $c_1(\boldsymbol{x})$ and $c_2(\boldsymbol{x})$ and send messages $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ to the other decoders. In the interactive procedures, Decoder 1 forms the distribution $p_1(\boldsymbol{x}; \boldsymbol{\zeta}_1)$, in which the nonlinear effect other than $c_1(\boldsymbol{x})$ (that is, $c_2(\boldsymbol{x})$ in the turbo decoding case of $K = 2$) is replaced by the estimate $\boldsymbol{\zeta}_1 (= \boldsymbol{\xi}_2)$, which is equal to the message $\boldsymbol{\zeta}_1$ sent from Decoder 2. In the general case of $K > 2$, $\boldsymbol{\zeta}_1$ summarizes all the messages, $\boldsymbol{\xi}_2, \cdots, \boldsymbol{\xi}_K$, from the other decoders. The same explanation holds for Decoder 2. The total linear estimate is given

by $\boldsymbol{\theta} \cdot \boldsymbol{x} = \boldsymbol{\xi}_1 \cdot \boldsymbol{x} + \boldsymbol{\xi}_2 \cdot \boldsymbol{x}$. The idea of turbo decoding is schematically shown in Fig.4. The projected distribution is written as

$$p_0(\boldsymbol{x}; \boldsymbol{\theta}) = \exp(c_0(\boldsymbol{x}) + \boldsymbol{\theta} \cdot \boldsymbol{x} - \varphi_0(\boldsymbol{\theta})) = \exp(c_0(\boldsymbol{x}) + \boldsymbol{\xi}_1 \cdot \boldsymbol{x} + \boldsymbol{\xi}_2 \cdot \boldsymbol{x} - \varphi_0(\boldsymbol{\theta})).$$

B. Equilibrium of Turbo Decoding

Assume that the decoding algorithm converges to a distribution $p_0(\boldsymbol{x}; \boldsymbol{\theta}^*)$, where $*$ is used to denote the equilibrium point. The distribution $p_0(\boldsymbol{x}; \boldsymbol{\theta}^*)$ is the approximation of the m–projection of $q(\boldsymbol{x})$ to M_0. The estimated parameter $\boldsymbol{\theta}^*$ satisfies $\boldsymbol{\theta}^* = \pi_{M_0} \circ p_1(\boldsymbol{x}; \boldsymbol{\zeta}_1^*) = \pi_{M_0} \circ p_2(\boldsymbol{x}; \boldsymbol{\zeta}_2^*)$ and $\boldsymbol{\theta}^* = \boldsymbol{\xi}_1^* + \boldsymbol{\xi}_2^* = \boldsymbol{\zeta}_1^* + \boldsymbol{\zeta}_2^*$ from the definition of the algorithm.

The converged distributions $p_1(\boldsymbol{x}; \boldsymbol{\zeta}_1^*)$, $p_2(\boldsymbol{x}; \boldsymbol{\zeta}_2^*)$, and $p_0(\boldsymbol{x}; \boldsymbol{\theta}^*)$ satisfy two conditions:

$$1) \quad \pi_{M_0} \circ p_1(\boldsymbol{x}; \boldsymbol{\zeta}_1^*) = \pi_{M_0} \circ p_2(\boldsymbol{x}; \boldsymbol{\zeta}_2^*) = \boldsymbol{\theta}^* \tag{13}$$

$$2) \quad \boldsymbol{\theta}^* = \boldsymbol{\xi}_1^* + \boldsymbol{\xi}_2^* = \boldsymbol{\zeta}_1^* + \boldsymbol{\zeta}_2^*. \tag{14}$$

The first condition can be rewritten with the expectation parameter defined in eqs.(9) and (10) as

$$\boldsymbol{\eta}_1(\boldsymbol{\zeta}_1^*) = \boldsymbol{\eta}_2(\boldsymbol{\zeta}_2^*) = \boldsymbol{\eta}_0(\boldsymbol{\theta}^*).$$

In order to give an information geometrical view of these conditions, we define two submanifolds in S. The first is the m–flat submanifold, $M(\boldsymbol{\theta})$, which we call the equimarginal submanifold, attached to each $p_0(\boldsymbol{x}; \boldsymbol{\theta}) \in M_0$. It is defined by

$$M(\boldsymbol{\theta}) = \left\{ p(\boldsymbol{x}) \;\middle|\; p(\boldsymbol{x}) \in S, \; \sum_{\boldsymbol{x}} p(\boldsymbol{x}) \boldsymbol{x} = \sum_{\boldsymbol{x}} p_0(\boldsymbol{x}; \boldsymbol{\theta}) \boldsymbol{x} = \boldsymbol{\eta}_0(\boldsymbol{\theta}) \right\}.$$

The expectation of $\boldsymbol{x}$ is equal to $\boldsymbol{\eta}_0(\boldsymbol{\theta})$ for any $p(\boldsymbol{x}) \in M(\boldsymbol{\theta})$. Hence, the m–projection of any $p(\boldsymbol{x}) \in M(\boldsymbol{\theta})$ to M_0 coincides with $p_0(\boldsymbol{x}; \boldsymbol{\theta})$. In other words, $M(\boldsymbol{\theta})$ is the inverse image of $p_0(\boldsymbol{x}; \boldsymbol{\theta})$ of the m-projection. For every $\boldsymbol{\theta}$, there exist unique $p_1(\boldsymbol{x}; \boldsymbol{\zeta}_1) \in M_1$ and $p_2(\boldsymbol{x}; \boldsymbol{\zeta}_2) \in M_2$ such that the expectations of $\boldsymbol{x}$ with respect to $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)$ and $p_0(\boldsymbol{x}; \boldsymbol{\theta})$ satisfy

$$\boldsymbol{\eta}_1(\boldsymbol{\zeta}_1) = \boldsymbol{\eta}_2(\boldsymbol{\zeta}_2) = \boldsymbol{\eta}_0(\boldsymbol{\theta}).$$

We denote the parameters that satisfy this equation by $\boldsymbol{\zeta}_1(\boldsymbol{\theta})$ and $\boldsymbol{\zeta}_2(\boldsymbol{\theta})$. In other words, we define $\boldsymbol{\zeta}_1(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \pi_{M_1} \circ p_0(\boldsymbol{x}; \boldsymbol{\theta})$ and $\boldsymbol{\zeta}_2(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \pi_{M_2} \circ p_0(\boldsymbol{x}; \boldsymbol{\theta})$. Obviously, $p_1(\boldsymbol{x}; \boldsymbol{\zeta}_1(\boldsymbol{\theta})), p_2(\boldsymbol{x}; \boldsymbol{\zeta}_2(\boldsymbol{\theta})) \in M(\boldsymbol{\theta})$, and $\pi_{M_0} \circ p_1(\boldsymbol{x}; \boldsymbol{\zeta}_1(\boldsymbol{\theta})) = \pi_{M_0} \circ p_2(\boldsymbol{x}; \boldsymbol{\zeta}_2(\boldsymbol{\theta}))) = \boldsymbol{\theta}$; however generally, $\boldsymbol{\zeta}_1(\boldsymbol{\theta}) + \boldsymbol{\zeta}_2(\boldsymbol{\theta}) \neq \boldsymbol{\theta}$ except for equilibrium point $\boldsymbol{\theta}^*$. The projection theorem shows that $M(\boldsymbol{\theta})$ is orthogonal to M_0, M_1, and M_2 (Fig.5), and that $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r(\boldsymbol{\theta}))$ is the intersection of M_r and $M(\boldsymbol{\theta})$.

We next define an e–flat submanifold $E(\boldsymbol{\theta})$ connecting $p_0(\boldsymbol{x}; \boldsymbol{\theta})$, $p_1(\boldsymbol{x}; \boldsymbol{\zeta}_1(\boldsymbol{\theta}))$, and $p_2(\boldsymbol{x}; \boldsymbol{\zeta}_2(\boldsymbol{\theta}))$ in a log-linear manner:

$$E(\boldsymbol{\theta}) = \left\{ p(\boldsymbol{x}) = C p_0(\boldsymbol{x}; \boldsymbol{\theta})^{t_0} p_1(\boldsymbol{x}; \boldsymbol{\zeta}_1(\boldsymbol{\theta}))^{t_1} p_2(\boldsymbol{x}; \boldsymbol{\zeta}_2(\boldsymbol{\theta}))^{t_2} \;\middle|\; \sum_{r=0}^{2} t_r = 1 \right\}, \;\; C : \text{normalization factor.}$$

This manifold is a two-dimensional e–affine subspace of S. Apparently, $p_0(\boldsymbol{x};\boldsymbol{\theta})$, $p_1(\boldsymbol{x};\boldsymbol{\zeta}_1(\boldsymbol{\theta}))$, and $p_2(\boldsymbol{x};\boldsymbol{\zeta}_2(\boldsymbol{\theta}))$ belongs to $E(\boldsymbol{\theta})$. Moreover, at equilibrium $\boldsymbol{\theta}^*$, $q(\boldsymbol{x})$ is included in $E(\boldsymbol{\theta}^*)$. This is easily proved by setting $t_0 = -1$, $t_1 = t_2 = 1$, and eq.(14)

$$\begin{aligned} C\frac{p_1(\boldsymbol{x};\boldsymbol{\zeta}_1^*)p_2(\boldsymbol{x};\boldsymbol{\zeta}_2^*)}{p_0(\boldsymbol{x};\boldsymbol{\theta}^*)} =& C\exp(2c_0(\boldsymbol{x}) + c_1(\boldsymbol{x}) + c_2(\boldsymbol{x}) + (\boldsymbol{\zeta}_1^* + \boldsymbol{\zeta}_2^*)\cdot\boldsymbol{x} - (c_0(\boldsymbol{x}) + \boldsymbol{\theta}^*\cdot\boldsymbol{x})) \\ =& C\exp(c_0(\boldsymbol{x}) + c_1(\boldsymbol{x}) + c_2(\boldsymbol{x})) = q(\boldsymbol{x}). \end{aligned}$$

This discussion is summarized in the following theorem.

Theorem 3: At the equilibrium of the turbo decoding algorithm, $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$, $p_1(\boldsymbol{x};\boldsymbol{\zeta}_1^*)$, and $p_2(\boldsymbol{x};\boldsymbol{\zeta}_2^*)$ belong to the equimarginal submanifold $M(\boldsymbol{\theta}^*)$: its e–flat version, $E(\boldsymbol{\theta}^*)$, includes $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$, $p_1(\boldsymbol{x};\boldsymbol{\zeta}_1^*)$, $p_2(\boldsymbol{x};\boldsymbol{\zeta}_2^*)$, and $q(\boldsymbol{x})$.

The theorem shows the information geometrical structure of the equilibrium point. If $M(\boldsymbol{\theta}^*)$ includes $q(\boldsymbol{x})$, $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$ gives MPM decoding based on $q(\boldsymbol{x})$, since the soft decoding of $q(\boldsymbol{x})$ is equivalent to the m–projection of $q(\boldsymbol{x})$ to M_0, and $M(\boldsymbol{\theta}^*)$ is orthogonal to M_0 at $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$. However, since the m–flatness and the e–flatness do not coincide in general, $M(\boldsymbol{\theta}^*)$ does not necessarily include $q(\boldsymbol{x})$, while its e–flat version, $E(\boldsymbol{\theta}^*)$, includes $q(\boldsymbol{x})$ instead of $M(\boldsymbol{\theta}^*)$. This shows that turbo decoding approximates MPM decoding by replacing the m–flat manifold $M(\boldsymbol{\theta}^*)$ with the e–flat manifold $E(\boldsymbol{\theta}^*)$. It should be noted that $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$ is not the e–projection of $q(\boldsymbol{x})$ to M_0 either, because $E(\boldsymbol{\theta}^*)$ is not necessarily orthogonal to M_0. When it is orthogonal, it minimizes the KL–divergence $D[p_0(\boldsymbol{x};\boldsymbol{\theta}); q(\boldsymbol{x})]$, $\boldsymbol{\theta} \in \mathcal{R}^N$, which gives the naive mean field approximation [15]. The replacement of m–projection with e–projection shares the similar idea of the mean field approximation[9], [15], [16], [17], [18]. Generally, there is a discrepancy between $M(\boldsymbol{\theta}^*)$ and $E(\boldsymbol{\theta}^*)$, which causes a decoding error (Fig.6). This suggests a possibility of a new method to improve the iterative decoding. We will study this in section VII.

C. Local Stability Analysis of Equilibrium Point

We discuss the local stability condition in this subsection. Let $I_0(\boldsymbol{\theta})$ be the Fisher information matrix of $p_0(\boldsymbol{x};\boldsymbol{\theta})$, and $I_r(\boldsymbol{\zeta}_r)$ be that of $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)$, $r = 1, 2$. Since they belong to the exponential family, we have the following relations:

$$\begin{aligned} I_0(\boldsymbol{\theta}) &= \partial_{\boldsymbol{\theta}\boldsymbol{\theta}}\varphi_0(\boldsymbol{\theta}) = \partial_{\boldsymbol{\theta}}\boldsymbol{\eta}_0(\boldsymbol{\theta}), \\ I_r(\boldsymbol{\zeta}_r) &= \partial_{\boldsymbol{\zeta}_r\boldsymbol{\zeta}_r}\varphi_r(\boldsymbol{\zeta}_r) = \partial_{\boldsymbol{\zeta}_r}\boldsymbol{\eta}_r(\boldsymbol{\zeta}_r), \quad r = 1, 2. \end{aligned}$$

Note that $I_0(\boldsymbol{\theta})$ is a diagonal matrix whose diagonal elements are

$$[I_0(\boldsymbol{\theta})]_{ii} = 1 - \eta_{0,i}^2.$$

In order to discuss the local stability, we give a sufficiently small perturbation, $\Delta\boldsymbol{\zeta}_2$, to $\boldsymbol{\zeta}_2^*$ and apply one step of the decoding procedure. Let $\boldsymbol{\zeta}_2' = \boldsymbol{\zeta}_2^* + \Delta\boldsymbol{\zeta}_2'$ be the parameter after one step of turbo decoding. From step 2, we have $\boldsymbol{\theta}^* + \Delta\boldsymbol{\theta} = \pi_{M_0}\circ p_2(\boldsymbol{x};\boldsymbol{\zeta}_2^* + \Delta\boldsymbol{\zeta}_2)$, such that

$$\boldsymbol{\eta}_0(\boldsymbol{\theta}^* + \Delta\boldsymbol{\theta}) = \boldsymbol{\eta}_2(\boldsymbol{\zeta}_2^* + \Delta\boldsymbol{\zeta}_2).$$

By simple expansion, we have

$$\boldsymbol{\eta}_0(\boldsymbol{\theta}^*) + I_0(\boldsymbol{\theta}^*)\Delta\boldsymbol{\theta} = \boldsymbol{\eta}_2(\boldsymbol{\zeta}_2^*) + I_2(\boldsymbol{\zeta}_2^*)\Delta\boldsymbol{\zeta}_2$$
$$\Delta\boldsymbol{\theta} = I_0(\boldsymbol{\theta}^*)^{-1} I_2(\boldsymbol{\zeta}_2^*)\Delta\boldsymbol{\zeta}_2.$$

Thus, $\boldsymbol{\zeta}_1$ in step 2 becomes

$$\boldsymbol{\zeta}_1 = \boldsymbol{\zeta}_1^* + (I_0(\boldsymbol{\theta}^*)^{-1} I_2(\boldsymbol{\zeta}_2^*) - E_N)\Delta\boldsymbol{\zeta}_2.$$

Following the same line for step 3, $\Delta\boldsymbol{\zeta}_2'$ is given by

$$\Delta\boldsymbol{\zeta}_2' = (I_0(\boldsymbol{\theta}^*)^{-1} I_1(\boldsymbol{\zeta}_1^*) - E_N)(I_0(\boldsymbol{\theta}^*)^{-1} I_2(\boldsymbol{\zeta}_2^*) - E_N)\Delta\boldsymbol{\zeta}_2 = \mathcal{T}_{turbo}\Delta\boldsymbol{\zeta}_2,$$

where

$$\mathcal{T}_{turbo} = (I_0(\boldsymbol{\theta}^*)^{-1} I_1(\boldsymbol{\zeta}_1^*) - E_N)(I_0(\boldsymbol{\theta}^*)^{-1} I_2(\boldsymbol{\zeta}_2^*) - E_N).$$

This shows that original perturbation $\Delta\boldsymbol{\zeta}_2$ becomes $\mathcal{T}_{turbo}\Delta\boldsymbol{\zeta}_2$ after one iteration.

Theorem 4: When $|\lambda_i| < 1$ for all i, where λ_i are the eigenvalues of matrix $\mathcal{T}_{turbo}$, the equilibrium point is locally stable.

This theorem coincides with the result of Richardson[2], Sect. V-A.

VI. Information Geometry of LDPC Decoding

A. Information Geometry of Decoding Process

The LDPC decoding procedure in subsection II-B is rewritten in the information geometrical framework as follows.

LDPC decoding (information geometrical view)

Initialization: For $t = 0$, set $\boldsymbol{\zeta}_r^0 = 0$ $(r = 1, \cdots, K)$. For $t = 0, 1, 2, \cdots$, compose $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^t) \in M_r$.

Horizontal step: Calculate the $m-$projection of $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^t)$ to M_0 and define $\boldsymbol{\xi}_r^{t+1}$ as

$$\boldsymbol{\xi}_r^{t+1} = \pi_{M_0} \circ p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^t) - \boldsymbol{\zeta}_r^t, \quad r = 1, \cdots, K. \tag{15}$$

Vertical step: Update $\{\boldsymbol{\zeta}_r^{t+1}\}$ and $\boldsymbol{\theta}^{t+1}$:

$$\boldsymbol{\theta}^{t+1} = \sum_{r=1}^{K} \boldsymbol{\xi}_r^{t+1}, \quad \boldsymbol{\zeta}_r^{t+1} = \boldsymbol{\theta}^{t+1} - \boldsymbol{\xi}_r^{t+1}, \ \ r = 1, \cdots, K.$$

Convergence: If $\boldsymbol{\theta}^t$ does not converge, repeat the process by incrementing t by 1.

Here, $\boldsymbol{\xi}_r$ is a message from each decoder that expresses the contribution of $c_r(\boldsymbol{x})$, and $\boldsymbol{\theta}$ integrates all the messages. Each decoder summarizes the information from all the other decoders in the form of the prior $\omega(\boldsymbol{x}; \boldsymbol{\zeta}_r)$. For turbo

decoding, K is equal to 2, and $\boldsymbol{\xi}_1 = \boldsymbol{\zeta}_2$ and $\boldsymbol{\xi}_2 = \boldsymbol{\zeta}_1$. Therefore, eq.(11) and eq.(12) are both equivalent to eq.(15) in LDPC decoding. The main difference between turbo and LDPC decoding is that the turbo decoding updates $\boldsymbol{\zeta}_r$ sequentially, while LDPC decoding updates them simultaneously.

B. Equilibrium and Stability

The equilibrium of LDPC codes satisfies two conditions:

$$1) \quad \pi_{M_0} \circ p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^*) = \boldsymbol{\theta}^*, \ \ r = 1, \cdots, K.$$

which can be rewritten with the expectation parameters as,

$$\boldsymbol{\eta}_0(\boldsymbol{\theta}^*) = \boldsymbol{\eta}_1(\boldsymbol{\zeta}_1^*) = \cdots = \boldsymbol{\eta}_K(\boldsymbol{\zeta}_K^*).$$

$$2) \quad \boldsymbol{\theta}^* = \sum_{r=1}^{K} \boldsymbol{\xi}_r^* = \frac{1}{K-1} \sum_{r=1}^{K} \boldsymbol{\zeta}_r^*.$$

Theorem 3 holds for LDPC decoding, in which the definitions of submanifold $E(\boldsymbol{\theta})$ must be extended as follows:

$$E(\boldsymbol{\theta}) = \left\{ p(\boldsymbol{x}) \ \middle| \ p(\boldsymbol{x}) = C p_0(\boldsymbol{x}; \boldsymbol{\theta})^{t_0} \prod_{r=1}^{K} p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r(\boldsymbol{\theta}))^{t_r}, \ \ t_r \in R, \ \sum_{r=0}^{K} t_r = 1 \right\}$$

C : normalization factor,

where $\boldsymbol{\zeta}_r(\boldsymbol{\theta})$ is defined as

$$\boldsymbol{\zeta}_r(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \pi_{M_r} \circ p_0(\boldsymbol{x}; \boldsymbol{\theta}), \quad r = 1, \cdots, K.$$

At the converged point, $q(\boldsymbol{x})$ is included in $E(\boldsymbol{\theta}^*)$, which can be proved by setting $t_0 = -(K-1), t_1 = t_2 = \cdots = 1$:

$$\begin{aligned}
& C \frac{\prod_{r=1}^{K} p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^*)}{p_0(\boldsymbol{x}; \boldsymbol{\theta}^*)^{K-1}} \\
&= C \exp\Big(K c_0(\boldsymbol{x}) + \sum_{r=1}^{K} c_r(\boldsymbol{x}) + \sum_{r=1}^{K} \boldsymbol{\zeta}_r^* \cdot \boldsymbol{x} - (K-1) c_0(\boldsymbol{x}) - (K-1) \boldsymbol{\theta}^* \cdot \boldsymbol{x} \Big) \\
&= C \exp(c_0(\boldsymbol{x}) + c_1(\boldsymbol{x}) + \cdots + c_K(\boldsymbol{x})) = q(\boldsymbol{x}).
\end{aligned}$$

This above equation proves that Theorem 3 holds for LDPC decoding.

We next show the local stability condition for LDPC decoding. Consider a case in which a sufficiently small perturbation is added to the equilibrium: $\boldsymbol{\zeta}_r = \boldsymbol{\zeta}_r^* + \Delta\boldsymbol{\zeta}_r$. The next state after a vertical step and a horizontal step is denoted by $\boldsymbol{\zeta}_r' = \boldsymbol{\zeta}_r^* + \Delta\boldsymbol{\zeta}_r'$. After the perturbation is added, the vertical step gives $\boldsymbol{\xi}_r = \boldsymbol{\xi}_r^* + \Delta\boldsymbol{\xi}_r$, where

$$\Delta\boldsymbol{\xi}_r = I_0(\boldsymbol{\xi}^*)^{-1} I_r(\boldsymbol{\zeta}_r^*) \Delta\boldsymbol{\zeta}_r - \Delta\boldsymbol{\zeta}_r = (I_0(\boldsymbol{\theta}^*)^{-1} I_r(\boldsymbol{\zeta}_r^*) - E_N) \Delta\boldsymbol{\zeta}_r.$$

Following the horizontal step, we have

$$\Delta\boldsymbol{\zeta}_r' = \sum_{r \neq s}^{K} (I_0(\boldsymbol{\theta}^*)^{-1} I_s(\boldsymbol{\zeta}_s^*) - E_N) \Delta\boldsymbol{\zeta}_s.$$

The local stability condition of LDPC decoding is summarized as follows.

Theorem 5: The linearization of the dynamics of LDPC decoding around the equilibrium is

$$\begin{pmatrix} \Delta\boldsymbol{\zeta}_1' \\ \vdots \\ \Delta\boldsymbol{\zeta}_K' \end{pmatrix} = \mathcal{T}_{LDPC} \begin{pmatrix} \Delta\boldsymbol{\zeta}_1 \\ \vdots \\ \Delta\boldsymbol{\zeta}_K \end{pmatrix},$$

where

$$\mathcal{T}_{LDPC} = \begin{pmatrix} O & I_0^{-1}I_2 - E_N & \cdots & I_0^{-1}I_K - E_N \\ I_0^{-1}I_1 - E_N & O & & \vdots \\ \vdots & & \ddots & \vdots \\ I_0^{-1}I_1 - E_N & \cdots & \cdots & O \end{pmatrix},$$

$I_0 = I_0(\boldsymbol{\theta}^*)$, and $I_r = I_r(\boldsymbol{\zeta}_r^*)$. The equilibrium is locally stable when every eigenvalue, λ_i $(i = 1, \cdots, NK)$, of $\mathcal{T}_{LDPC}$ satisfies $|\lambda_i| < 1$.

The local stability condition generally depends on the syndrome vector, $\tilde{\boldsymbol{y}}$. However, when the partial conditional probability, $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^*)$, is close to $p_0(\boldsymbol{x}; \boldsymbol{\theta}^*)$, $I_r \approx I_0$. For LDPC, $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^*) \approx p_0(\boldsymbol{x}; \boldsymbol{\theta}^*)$ because of the sparsity of the parity check matrix. This implies that all the eigenvalues of $\mathcal{T}_{LDPC}$ should be small, which leads to a stable and quick convergence.

VII. Analysis of Decoding Errors

A. Framework of Error Analysis

We have described the information geometrical framework of the decoding algorithms and have shown how MPM decoding is approximated by these decoding algorithms. In this section we analyze the goodness of the approximation and give a correction term for improving the approximation. We also provide an explanation why the sparsity, i.e., low density, of the parity check matrix has an advantage.

For the following discussion, we define the family of distributions,

$$M_S = \{p(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{v})\},$$

by using two sets of parameters: $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_N)^T \in \mathcal{R}^N$ and $\boldsymbol{v} = (v_1, \ldots, v_K)^T \in \mathcal{R}^K$.

$$\begin{aligned} p(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{v}) &= \exp\Big(c_0(\boldsymbol{x}) + \boldsymbol{\theta} \cdot \boldsymbol{x} + \sum_{r=1}^{K} v_r c_r(\boldsymbol{x}) - \varphi(\boldsymbol{\theta}, \boldsymbol{v})\Big) \\ &= \exp(c_0(\boldsymbol{x}) + \boldsymbol{\theta} \cdot \boldsymbol{x} + \boldsymbol{v} \cdot \boldsymbol{c}(\boldsymbol{x}) - \varphi(\boldsymbol{\theta}, \boldsymbol{v})) \\ \varphi(\boldsymbol{\theta}, \boldsymbol{v}) &\stackrel{\text{def}}{=} \ln \sum_{\boldsymbol{x}} \exp(c_0(\boldsymbol{x}) + \boldsymbol{\theta} \cdot \boldsymbol{x} + \boldsymbol{v} \cdot \boldsymbol{c}(\boldsymbol{x})), \quad \boldsymbol{c}(\boldsymbol{x}) \stackrel{\text{def}}{=} (c_1(\boldsymbol{x}), \ldots, c_K(\boldsymbol{x}))^T. \end{aligned}$$

The family M_S is a $(K + N)$–dimensional exponential family. The manifolds $M_0 = \{p_0(\boldsymbol{x}; \boldsymbol{\theta})\}$ and $M_r = \{p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)\}$ are submanifolds of M_S since $M_0 = \{p(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{v}) | \boldsymbol{v} = \mathbf{o}\}$ and $M_r = \{p(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{v}) | \boldsymbol{v} = \boldsymbol{e}_r\}$, where $\boldsymbol{e}_r$ is the unit vector

$$\boldsymbol{e}_r = (0, \ldots, 0, \underset{\hat{r}}{1}, 0, \ldots, 0)^T.$$

It also includes $q(\boldsymbol{x})$, when we set $\boldsymbol{\theta} = \mathbf{o}$ and $\boldsymbol{v} = \mathbf{1}_K$:

$$\mathbf{1}_K = (\underbrace{1, \cdots, 1}_{K})^T = \sum_{r=1}^{K} \boldsymbol{e}_r.$$

We denote the expectation parameter of $p(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{v}) \in M_S$ by $\boldsymbol{\eta}(\boldsymbol{\theta}, \boldsymbol{v}) = (\eta_1(\boldsymbol{\theta}, \boldsymbol{v}), \ldots, \eta_N(\boldsymbol{\theta}, \boldsymbol{v}))^T$, which is given by

$$\boldsymbol{\eta}(\boldsymbol{\theta}, \boldsymbol{v}) = \partial_{\boldsymbol{\theta}} \varphi(\boldsymbol{\theta}, \boldsymbol{v}) = \sum_{\boldsymbol{x}} p(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{v}) \boldsymbol{x}.$$

B. Analysis of Equimarginal Submanifold $M(\boldsymbol{\theta}^*)$

Let $p(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{v})$ be the distributions included in the equimarginal submanifold, $M(\boldsymbol{\theta}^*)$; that is, $\boldsymbol{\eta}(\boldsymbol{\theta}, \boldsymbol{v}) = \boldsymbol{\eta}(\boldsymbol{\theta}^*, \mathbf{o}) \stackrel{\text{def}}{=} \boldsymbol{\eta}(\boldsymbol{\theta}^*)$. This constraint makes $\boldsymbol{\theta}$ an implicit function of $\boldsymbol{v}$, which is denoted by $\boldsymbol{\theta}(\boldsymbol{v})$. Note that $\boldsymbol{\theta}^* = \boldsymbol{\theta}(\mathbf{o})$. More precisely,

$$\boldsymbol{\eta}(\boldsymbol{\theta}(\boldsymbol{v}), \boldsymbol{v}) = \boldsymbol{\eta}(\boldsymbol{\theta}(\mathbf{o}), \mathbf{o}) = \boldsymbol{\eta}(\boldsymbol{\theta}^*),$$

for any $\boldsymbol{v}$. We analyze how $\boldsymbol{\theta}$ changes from $\boldsymbol{\theta}^*$ as $\boldsymbol{v}$ changes from $\mathbf{o}$ and finally becomes $\mathbf{1}_K$. We start by introducing the derivative $D/\partial \boldsymbol{v}$ along $M(\boldsymbol{\theta})$:

$$\mathbf{o} = \frac{D}{\partial \boldsymbol{v}} \boldsymbol{\eta}(\boldsymbol{\theta}, \boldsymbol{v}) = \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{v}} + \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{v}}. \tag{16}$$

The structural quantities $\partial \boldsymbol{\eta}/\partial \boldsymbol{\theta}$ and $\partial \boldsymbol{\eta}/\partial \boldsymbol{v}$ are the Fisher information because $\boldsymbol{\eta} = \partial \varphi(\boldsymbol{\theta}, \boldsymbol{v})/\partial \boldsymbol{\theta}$. We use the index notation in which suffixes i, j, and k are for $\boldsymbol{\theta}$ and r, s, and t are for $\boldsymbol{v}$. In component form, $(I_0(\boldsymbol{\theta}) =) G_{\boldsymbol{\theta}\boldsymbol{\theta}} \stackrel{\text{def}}{=} (\partial \boldsymbol{\eta}/\partial \boldsymbol{\theta})$ and $G_{\boldsymbol{\theta}\boldsymbol{v}} \stackrel{\text{def}}{=} (\partial \boldsymbol{\eta}/\partial \boldsymbol{v})$ are defined as

$$g_{ij}(\boldsymbol{\theta}) = \frac{\partial \eta_i}{\partial \theta_j} = G_{ij}(\boldsymbol{\theta}) = I_{0,ij}(\boldsymbol{\theta}), \quad g_{ir}(\boldsymbol{\theta}) = \frac{\partial \eta_i}{\partial v_r}.$$

From eq.(16), $G_{\boldsymbol{\theta}\boldsymbol{\theta}}$, and $G_{\boldsymbol{\theta}\boldsymbol{v}}$ we have

$$\begin{aligned} \mathbf{o} &= I_0(\boldsymbol{\theta}) \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{v}} + G_{\boldsymbol{\theta}\boldsymbol{v}}(\boldsymbol{\theta}) \\ \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{v}} &= -I_0^{-1}(\boldsymbol{\theta}) G_{\boldsymbol{\theta}\boldsymbol{v}}(\boldsymbol{\theta}) \stackrel{\text{def}}{=} -\tilde{G}_{\boldsymbol{\theta}\boldsymbol{v}}. \end{aligned} \tag{17}$$

Similarly, from

$$\frac{D^2}{\partial \boldsymbol{v} \partial \boldsymbol{v}} \boldsymbol{\eta}(\boldsymbol{\theta}, \boldsymbol{v}) = 0,$$

we have

$$I_0(\boldsymbol{\theta}) \frac{\partial^2 \boldsymbol{\theta}}{\partial \boldsymbol{v} \partial \boldsymbol{v}} = -T_{\boldsymbol{\theta}\boldsymbol{v}\boldsymbol{v}} - T_{\boldsymbol{\theta}\boldsymbol{\theta}\boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{v}} \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{v}} - 2 T_{\boldsymbol{\theta}\boldsymbol{\theta}\boldsymbol{v}} \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{v}}, \tag{18}$$

where

$$T_{\boldsymbol{\theta}\boldsymbol{\theta}\boldsymbol{\theta}} = \frac{\partial^3 \varphi}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta} \partial \boldsymbol{\theta}}, \quad T_{\boldsymbol{\theta}\boldsymbol{\theta}\boldsymbol{v}} = \frac{\partial^3 \varphi}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta} \partial \boldsymbol{v}}, \quad T_{\boldsymbol{\theta}\boldsymbol{v}\boldsymbol{v}} = \frac{\partial^3 \varphi}{\partial \boldsymbol{\theta} \partial \boldsymbol{v} \partial \boldsymbol{v}}.$$

More explicitly, by using the index notation, we have

$$\sum_j g_{ij}\frac{\partial^2\theta_j}{\partial v_r\partial v_s} = -T_{irs} - \sum_{j,k} T_{ijk}\frac{\partial\theta_j}{\partial v_r}\frac{\partial\theta_k}{\partial v_s} - 2\sum_j T_{ijr}\frac{\partial\theta_j}{\partial v_s}.$$

By replacing $\partial\boldsymbol{\theta}/\partial\boldsymbol{v}$ in eq.(18) with the result of eq.(17), we get

$$I_0(\boldsymbol{\theta})\frac{\partial^2\boldsymbol{\theta}}{\partial\boldsymbol{v}\partial\boldsymbol{v}} = -T_{\boldsymbol{\theta vv}} - T_{\boldsymbol{\theta\theta\theta}}\tilde{G}_{\boldsymbol{\theta v}}\tilde{G}_{\boldsymbol{\theta v}} + 2T_{\boldsymbol{\theta\theta v}}\tilde{G}_{\boldsymbol{\theta v}}.$$

The differential operator $D/d\boldsymbol{v}$ at $(\boldsymbol{\theta},\boldsymbol{v}) = (\boldsymbol{\theta}^*, \mathbf{o})$ is written as

$$\frac{D}{d\boldsymbol{v}}\Big|_{(\boldsymbol{\theta},\boldsymbol{v})=(\boldsymbol{\theta}^*,\mathbf{o})} \stackrel{\text{def}}{=} B = \frac{\partial}{\partial\boldsymbol{v}} - \tilde{G}_{\boldsymbol{\theta v}}(\boldsymbol{\theta}^*)\frac{\partial}{\partial\boldsymbol{\theta}},$$

In the component form, it is

$$B_r = \frac{\partial}{\partial v_r} - \sum_i \tilde{g}_{ir}(\boldsymbol{\theta}^*)\frac{\partial}{\partial\theta_i}.$$

Following some calculations, we have

$$\frac{\partial^2\boldsymbol{\theta}}{\partial\boldsymbol{v}\partial\boldsymbol{v}} = -I_0(\boldsymbol{\theta}^*)^{-1}B^2\boldsymbol{\eta}.$$

We denote the (r,s) component of B^2 by $B_{rs} = B_rB_s$. Note that $B^2\boldsymbol{\eta} \neq \mathbf{o}$ while $(D^2/\partial\boldsymbol{v}\partial\boldsymbol{v})\boldsymbol{\eta} = \mathbf{o}$.

The second-order approximation of $\boldsymbol{\theta}(\boldsymbol{v})$ around $(\boldsymbol{\theta}^*, \mathbf{o})$ is given by

$$\begin{aligned}\boldsymbol{\theta}(\boldsymbol{v}) =& \boldsymbol{\theta}^* + \frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{v}}\boldsymbol{v} + \frac{1}{2}\boldsymbol{v}^T\frac{\partial^2\boldsymbol{\theta}}{\partial\boldsymbol{v}\partial\boldsymbol{v}}\boldsymbol{v},\\ =& \boldsymbol{\theta}^* - \tilde{G}_{\boldsymbol{\theta v}}\boldsymbol{v} - \frac{1}{2}\boldsymbol{v}^T I_0^{-1}(\boldsymbol{\theta}^*)(B^2\boldsymbol{\eta})\boldsymbol{v},\end{aligned}$$

where the values of G and T are to be evaluated at $(\boldsymbol{\theta}^*, \mathbf{o})$. By plugging $\boldsymbol{v} = \mathbf{1}_K$ into the formula, we have

$$\theta_i(\mathbf{1}_K) = \theta_i^* - \sum_r \tilde{g}_{ir}(\boldsymbol{\theta}^*) - \frac{1}{2}(I_0^{-1}(\boldsymbol{\theta}^*))_{ii}\Big(\sum_{r,s} B_{rs}\Big)\eta_i(\boldsymbol{\theta}^*), \tag{19}$$

which shows the point at which $M(\boldsymbol{\theta}^*)$ intersects the submodels $\{p(\boldsymbol{x};\boldsymbol{\theta},\mathbf{1}_K)\}$. Since $q(\boldsymbol{x})$ is given by $p(\boldsymbol{x};\mathbf{o},\mathbf{1}_K)$, $\boldsymbol{\theta}(\mathbf{1}_K)$ is related to the discrepancy of $q(\boldsymbol{x})$ and the iterative decoding result.

This result is based on the perturbation analysis, of which justification is outlined below. When ϵ is small, the Taylor expansion for function $f(x)$ is

$$f(\epsilon) = f(0) + f'(0)\epsilon + \frac{1}{2}f''(0)\epsilon^2 + O(\epsilon^3).$$

When we rescale $v = x/\epsilon$,

$$f(v) = f(0) + \epsilon f'(0)v + \frac{1}{2}\epsilon^2 f''(0)v^2 + O(\epsilon^3).$$

In our analysis of iterative decoding, $x = \epsilon$ corresponds to $v_r = 1$, where the k-th derivative is of the order ϵ^k. We have assumed that the effects of $\boldsymbol{v}$ are small, and we take the expansion with respect to $\boldsymbol{v}$ in terms of ϵ. We finally set $\epsilon = 1$, and the results are valid in the above sense.

In order to conclude our analysis of the decoding error based on perturbation analysis, we consider two distributions:

$$p(\boldsymbol{x};\boldsymbol{\theta}^*,\mathbf{o}) = \exp(c_0(\boldsymbol{x}) + \boldsymbol{\theta}^* \cdot \boldsymbol{x} - \varphi(\boldsymbol{\theta}^*,\mathbf{o}))$$

$$p(\boldsymbol{x};\boldsymbol{\zeta}_r,\epsilon\boldsymbol{e}_r) = \exp(c_0(\boldsymbol{x}) + \boldsymbol{\zeta}_r \cdot \boldsymbol{x} + \epsilon c_r(\boldsymbol{x}) - \varphi(\boldsymbol{\zeta}_r,\epsilon\boldsymbol{e}_r)).$$

Note that $p(\boldsymbol{x};\boldsymbol{\theta}^*,\mathbf{o}) \equiv p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$, and $p(\boldsymbol{x};\boldsymbol{\zeta}_r,\epsilon\boldsymbol{e}_r)\big|_{\epsilon=1} = p_r(\boldsymbol{x};\boldsymbol{\zeta}_r^*)$. Let $p(\boldsymbol{x};\boldsymbol{\zeta}_r,\epsilon\boldsymbol{e}_r)$, $r = 1,\ldots,K$, be included in $M(\boldsymbol{\theta}^*)$. From the result of eq.(19), $\boldsymbol{\zeta}_r - \boldsymbol{\theta}^*$ is approximated in the power series of ϵ:

$$\boldsymbol{\zeta}_r - \boldsymbol{\theta}^* \simeq -\tilde{G}_{\boldsymbol{\theta v}}(\boldsymbol{\theta}^*)\boldsymbol{e}_r\epsilon - \frac{1}{2}I_0^{-1}(\boldsymbol{\theta}^*)B_{rr}\boldsymbol{\eta}(\boldsymbol{\theta}^*)\epsilon^2.$$

This gives the approximation of $\boldsymbol{\xi}_r^*$ as ϵ becomes 1:

$$\boldsymbol{\zeta}_r^* - \boldsymbol{\theta}^* = -\boldsymbol{\xi}_r^* \simeq -\tilde{G}_{\boldsymbol{\theta v}}(\boldsymbol{\theta}^*)\boldsymbol{e}_r - \frac{1}{2}I_0^{-1}(\boldsymbol{\theta}^*)B_{rr}\boldsymbol{\eta}(\boldsymbol{\theta}^*).$$

Hence, $\boldsymbol{\theta}^*$ satisfies

$$\boldsymbol{\theta}^* = \sum_r \boldsymbol{\xi}_r^* \simeq \tilde{G}_{\boldsymbol{\theta v}}(\boldsymbol{\theta}^*)\mathbf{1}_K + \frac{1}{2}I_0^{-1}(\boldsymbol{\theta}^*)\sum_r B_{rr}\boldsymbol{\eta}(\boldsymbol{\theta}^*). \tag{20}$$

Consider another distribution,

$$p(\boldsymbol{x};\boldsymbol{u},\epsilon\mathbf{1}_K) = \exp(c_0(\boldsymbol{x}) + \boldsymbol{u} \cdot \boldsymbol{x} + \epsilon\mathbf{1}_K \cdot \boldsymbol{c}(\boldsymbol{x}) - \varphi(\boldsymbol{u},\epsilon\mathbf{1}_K)).$$

Note that $p(\boldsymbol{x};\mathbf{o},\epsilon\mathbf{1}_K)\big|_{\epsilon=1} = q(\boldsymbol{x})$ and that $p(\boldsymbol{x};\boldsymbol{u},\epsilon\mathbf{1}_K)$ is included in $M(\boldsymbol{\theta}^*)$. As ϵ increases from 0 to 1, $\boldsymbol{u}$ becomes $\boldsymbol{u}^*$, and generally $\boldsymbol{u}^* \neq \mathbf{o}$, which means $q(\boldsymbol{x})$ is generally not included in $M(\boldsymbol{\theta}^*)$.

From the result of eq.(19), we have

$$\boldsymbol{u}^* - \boldsymbol{\theta}^* \simeq -\tilde{G}_{\boldsymbol{\theta v}}(\boldsymbol{\theta}^*)\mathbf{1}_K - \frac{1}{2}I_0^{-1}(\boldsymbol{\theta}^*)\sum_{r,s} B_{rs}\boldsymbol{\eta}(\boldsymbol{\theta}^*). \tag{21}$$

From eqs.(20) and (21), we have

$$\boldsymbol{u}^* \simeq -\frac{1}{2}I_0^{-1}(\boldsymbol{\theta}^*)\sum_{r\neq s} B_{rs}\boldsymbol{\eta}(\boldsymbol{\theta}^*).$$

From the Taylor expansion, we have

$$\boldsymbol{\eta}(\mathbf{o},\mathbf{1}_K) \simeq \boldsymbol{\eta}(\boldsymbol{u}^*,\mathbf{1}_K) - \nabla_{\boldsymbol{\theta}}\boldsymbol{\eta}(\boldsymbol{\theta}^*)\boldsymbol{u}^* = \boldsymbol{\eta}(\boldsymbol{\theta}^*) + \frac{1}{2}\sum_{r\neq s} B_{rs}\boldsymbol{\eta}(\boldsymbol{\theta}^*). \tag{22}$$

Note that $\boldsymbol{\eta}(\mathbf{o},\mathbf{1}_K)$ is the expectation of $\boldsymbol{x}$ with respect to $q(\boldsymbol{x})$ which is equivalent to soft decoding based on $q(\boldsymbol{x})$. Equation (22) shows the difference between the ultimate goal of the decoding and the result of iterative decoding.

We summarize the above analysis:

Theorem 6: Let $\boldsymbol{\eta}_{MPM} \stackrel{\text{def}}{=} \boldsymbol{\eta}(\mathbf{o},\mathbf{1}_K)$ be the expectation of $\boldsymbol{x}$ with respect to $q(\boldsymbol{x})$, and $\boldsymbol{\eta}(\boldsymbol{\theta}^*)$ be the expectation with respect to the distribution obtained by iterative decoding. Then, $\boldsymbol{\eta}_{MPM}$ is approximated by decoding result $\boldsymbol{\eta}(\boldsymbol{\theta}^*)$:

$$\boldsymbol{\eta}_{MPM} \simeq \boldsymbol{\eta}(\boldsymbol{\theta}^*) + \frac{1}{2}\sum_{r\neq s} B_{rs}\boldsymbol{\eta}(\boldsymbol{\theta}^*). \tag{23}$$

C. Remark on $B_{rs}\eta_i$

We remark here that the error term is related to the curvature of $M(\boldsymbol{\theta}^*)$ without giving details about the definition of the e– and m–curvatures. See Amari and Nagaoka[12] for the mathematical details. We have shown that $M(\boldsymbol{\theta})$ is m–flat. This implies that the embedding m–curvature tensor vanishes; that is,

$$H_{rs}^{(m)i} = \frac{D^2}{\partial v_r \partial v_s}\eta_i(\boldsymbol{v}) = 0.$$

On the other hand, $M(\boldsymbol{\theta})$ is not e–flat, so the embedding e–curvature is given by

$$H_{rs}^{(e)i} = \frac{D^2}{\partial v_r \partial v_s}\theta_i(\boldsymbol{v}).$$

Its covariant version is given by

$$H_{rs}^{(e)i} = B_{rs}\eta_i,$$

which shows that the error term is directly related to the e–curvature of $M(\boldsymbol{\theta}^*)$.

VIII. Improving Decoding Errors for LDPC codes

A. Structural Terms

The terms $B_{rs}\eta_i$ are given by structural tensors G and T at $p_0(\boldsymbol{x};\boldsymbol{\theta}) \in M_0$. For LDPC codes, they are given by

$$g_{ir} = E_{p_0}[(x_i - \eta_i)(c_r(\boldsymbol{x}) - \bar{c}_r)],\ \ T_{ijr} = E_{p_0}[(x_i - \eta_i)(x_j - \eta_j)(c_r(\boldsymbol{x}) - \bar{c}_r)],$$

where E_{p_0} denotes the expectation with respect to $p_0(\boldsymbol{x};\boldsymbol{\theta})$, and

$$\bar{c}_r = E_{p_0}[c_r(\boldsymbol{x})] = \rho\tilde{y}_r \prod_{j\in\mathcal{L}_r} \eta_j.$$

Because the x_i's are independent with respect to $p_0(\boldsymbol{x};\boldsymbol{\theta})$, the following relations hold and are used for further calculation:

$$E_{p_0}[x_i c_r(\boldsymbol{x})] = \begin{cases} \eta_i \bar{c}_r, & \text{when } i \notin \mathcal{L}_r, \\ \dfrac{1}{\eta_i}\bar{c}_r & \text{when } i \in \mathcal{L}_r, \end{cases}$$

$$E_{p_0}[c_r(\boldsymbol{x})c_s(\boldsymbol{x})] = \frac{1}{P_{rs}}\bar{c}_r\bar{c}_s,$$

where

$$P_{rs} = \begin{cases} \prod_{j\in\mathcal{L}_r\cap\mathcal{L}_s} \eta_j^2. & \text{when } \mathcal{L}_r \cap \mathcal{L}_s \neq \emptyset, \\ 1 & \text{when } \mathcal{L}_r \cap \mathcal{L}_s = \emptyset. \end{cases}$$

The explicit forms of G and T are given in Appendix II.

B. Algorithm to Calculate Correction Term

From the result of Theorem 6, the soft-decoded $\boldsymbol{\eta}^*$ is improved by

$$\boldsymbol{\eta}_{MPM} = \boldsymbol{\eta}(\boldsymbol{\theta}^*) + \frac{1}{2}\sum_{r\neq s} B_{rs}\boldsymbol{\eta}(\boldsymbol{\theta}^*).$$

By calculating $B_{rs}\eta_i$ for $(r \neq s)$, (see Appendix III), we give the algorithm to calculate correction term $B_{rs}\eta_i$ as follows.

---

1) Calculate

$$\bar{c}_r = E_{p_0}[c_r(\boldsymbol{x})].$$

2) Given i, search for the pair (r, s) which includes i, that is, $i \in \mathcal{L}_r$ and $i \in \mathcal{L}_s$. Calculate

$$B_{rs}\eta_i = 2\frac{1-\eta_i^2}{\eta_i}\bar{c}_r\bar{c}_s\sum_{j\neq i}\frac{1-\eta_j^2}{\eta_j^2}h_{jr}h_{js}. \tag{24}$$

3) Given i, search for the pair (r, s) such that $i \in \mathcal{L}_r$ and $i \notin \mathcal{L}_s$. Calculate

$$B_{rs}\eta_i = \bar{c}_r\bar{c}_s\frac{1-\eta_i^2}{\eta_i}\left(-\frac{1-P_{rs}}{P_{rs}} + \sum_j \frac{1-\eta_j^2}{\eta_j^2}h_{jr}h_{js}\right). \tag{25}$$

4) The correction term is given by summing up over all (r, s) in the above two cases.

The summation in eq.(24) runs over $j \in \mathcal{L}_r \cap \mathcal{L}_s \setminus i$, and that in eq.(25) runs over $j \in \mathcal{L}_r \cap \mathcal{L}_s$. Thus, when the parity-check matrix is designed such that, for any r and s,

$$h_{ir}h_{is} = 1$$

holds for at most one i, that is, any two columns of the parity-check matrix have at most one overlapping positions of 1, all the principal terms of the correction vanishes (Tanaka et al.[19]), which leads to the following theorem for the LDPC codes.

Theorem 7: The principal term of the decoding error vanishes when parity-check matrix H has no pair of columns with an overlap of 1 more than once.

It is believed[4] that the average probability of a decoding error is small, when any two columns of parity-check matrix H do not have an overlap of 1 more than once. Intuitively, this avoidance prevents loops with a length of 4 from appearing in the graphical representation. Results of many experiments indicate that short loops are harmful for iterative decoding; that is, they worsen the decoding errors. Our result in Theorem 7 analytically supports this indication: the principal term of the decoding error vanishes when the parity-check matrix is sparse and prepared so that there are no two columns with an overlap of 1 more than once. Loops longer than 4 do not contribute to the decoding error at least via the principal term (although they may have effects via higher order terms). Many

LDPC codes have been designed to satisfy this criterion (MacKay[4]). The analysis presented here can be extended in a straightforward manner to higher order perturbation analysis in order to quantify these effects.

It should be noted that our approach is different from the approach commonly used to analyze the properties of iterative decoders since we do not consider any *ensemble* of codes. A typical reasoning found in the literature (e.g., [20]) is first to consider an ensemble of random parity-check matrices and show that the probability (over the ensemble) of short loops in the associated graph decreases to zero as the codelength tends to infinity while the column and row weights are kept finite. This means that the behavior of iterative decoders for codes with longer loops is the same as that in the loop-free case. The statistical-mechanical approach to performance analysis of Gallager-type codes[21] also assumes random ensembles. Our analysis, on the other hand, does not assume ensembles but allows the evaluation of the performance of the iterative decoders with any *single instance* of a the parity-check matrix with a finite codelength.

IX. Discussion and Conclusion

We have discussed the mechanism of the iterative decoding algorithms from the information geometrical viewpoint. We built a framework for analyzing the algorithms and used it to reveal their basic properties.

The problem of iterative decoding is summarized as a unified problem of marginalizing the probability distribution $q(\boldsymbol{x})$ in eq.(2). This problem is common to the belief propagation for the loopy belief diagram in artificial intelligence[6] and the Bethé approximation in statistical physics. In all of them, the direct marginalization of $q(\boldsymbol{x})$ is intractable, and only the marginalization of partial distributions $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)$, $r = 1, \cdots, K$, in eq.(4), is possible.

The marginalization of $q(\boldsymbol{x})$ is approximated through iterative processes of adjusting $\{\boldsymbol{\zeta}_r\}$, marginalizing $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)$, and integrating them into the approximated parameter $\boldsymbol{\theta}$. Both decoding algorithms were redefined with the information geometrical terms, and the conditions of the equilibrium were derived. They revealed an intuitive information geometrical meaning of the equilibrium point, which is summarized in Theorem 3. In the information geometrical terms, the ideal goal is to have the cross section of M_0 and an m–flat submanifold $M(\boldsymbol{\theta})$ including $q(\boldsymbol{x})$: however, instead of $M(\boldsymbol{\theta})$, an e–flat manifold $E(\boldsymbol{\theta})$ is used to obtain the decoding result. A new perspective arose from the theorem: the discrepancy between $M(\boldsymbol{\theta})$ and $E(\boldsymbol{\theta})$ gives the decoding error.

The principal term of the discrepancy was obtained through perturbation analysis, which is summarized in Theorem 6. The decoding error was given in eq.(23), and the correction term gives a method for improving the existing decoding algorithms. Moreover, since the correction term strongly depends on the encoders, it gives a new suggestion for designing the codes. We have done the perturbation analysis up to the second order, and it is possible to extend it to higher order analysis in a straightforward fashion.

We also derived the local stability conditions in Theorems 4 and 5. Although Theorem 4 coincides with the results of Richardson[2], Theorem 5 presents a new result for the local stability condition of LDPC codes. The global convergence property is another issue[22] which is one of our future works.

The belief propagation algorithm is not directly connected to the gradient method of minimizing a cost function. It has been pointed out that the final result is at the critical point of the Bethé free energy[9], [13].

For $\boldsymbol{\zeta}_1, \cdots, \boldsymbol{\zeta}_K$, and $\boldsymbol{\theta}$, we define the following function of $\{\boldsymbol{\zeta}_r\}$ and $\boldsymbol{\theta}$:

$$\mathcal{F}(\{\boldsymbol{\zeta}_r\}, \boldsymbol{\theta}) \stackrel{\text{def}}{=} D[p_0(\boldsymbol{x}; \boldsymbol{\theta}); q(\boldsymbol{x})] - \sum_{r=1}^{K} D[p_0(\boldsymbol{x}; \boldsymbol{\theta}); p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)].$$

The first term is rewritten as

$$D[p_0(\boldsymbol{x}; \boldsymbol{\theta}); q(\boldsymbol{x})] = E_{p_0}[c_0(\boldsymbol{x})] + \boldsymbol{\theta} \cdot \boldsymbol{\eta}_0(\boldsymbol{\theta}) - \varphi_0(\boldsymbol{\theta}) - \Big(\sum_{r=0}^{K} E_{p_0}[c_r(\boldsymbol{x})] + \ln C \Big).$$

The second term is rewritten as

$$\begin{aligned} \sum_{r=1}^{K} D[p_0(\boldsymbol{x}; \boldsymbol{\theta}); p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)] =& K(E_{p_0}[c_0(\boldsymbol{x})] + \boldsymbol{\theta} \cdot \boldsymbol{\eta}_0(\boldsymbol{\theta}) - \varphi_0(\boldsymbol{\theta})) \\ & - \sum_{r=1}^{K} (E_{p_0}[c_0(\boldsymbol{x})] + E_{p_0}[c_r(\boldsymbol{x})] + \boldsymbol{\zeta}_r \cdot \boldsymbol{\eta}_0(\boldsymbol{\theta}) - \varphi_r(\boldsymbol{\zeta}_r)). \end{aligned}$$

These three equations give

$$\mathcal{F}(\{\boldsymbol{\zeta}_r\}, \boldsymbol{\theta}) = (K-1)\varphi_0(\boldsymbol{\theta}) - \sum_{r=1}^{K} \varphi_r(\boldsymbol{\zeta}_r) - \ln C + \sum \boldsymbol{\zeta}_r \cdot (\boldsymbol{\eta}_0(\boldsymbol{\theta}) - \boldsymbol{\eta}_r(\boldsymbol{\zeta}_r)).$$

Since $\ln C$ is a constant, we neglect it and redefine $\mathcal{F}(\{\boldsymbol{\zeta}_r\}; \boldsymbol{\theta})$:

$$\mathcal{F}(\{\boldsymbol{\zeta}_r\}, \boldsymbol{\theta}) = (K-1)\varphi_0(\boldsymbol{\theta}) - \sum_{r=1}^{K} \varphi_r(\boldsymbol{\zeta}_r) + \sum \boldsymbol{\zeta}_r \cdot (\boldsymbol{\eta}_0(\boldsymbol{\theta}) - \boldsymbol{\eta}_r(\boldsymbol{\zeta}_r)).$$

When the $p_0(\boldsymbol{x}; \boldsymbol{\theta}), p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r) \in M(\boldsymbol{\theta})$, the last term vanishes, and this function with constraint $\boldsymbol{\zeta}_r = \boldsymbol{\zeta}_r(\boldsymbol{\theta})$ or $\boldsymbol{\eta}_r(\boldsymbol{\zeta}_r) = \boldsymbol{\eta}_0(\boldsymbol{\theta})$ coincides with the free energy introduced by Kabashima and Saad[9] using the statistical physical method.

The advantage of the information geometrical framework lies in its generality. The framework is common not only to turbo and LDPC codes, but is also generally valid for the Bethé approximation, the belief propagation applied to a loopy belief diagram, and its variants such as TRP[23] and the CCCP algorithm[24]. We have to work for reformulation of the problem in different terms. Another important extension will be found when we use different models of channels. It is easy to extend the result for any memoryless channel (see Appendix I), and by employing such channels, we can derive wide varieties of the turbo and the LDPC type decoding algorithms.

This study is a first step towards information geometrical understanding of turbo and LDPC codes. By using the framework presented in this paper, we expect that further understanding will appear and new improvements will emerge.

APPENDIX I
EXTENSION TO GENERAL MEMORYLESS CHANNEL

The information geometrical framework in this paper can be easily extended to the case where the channel is a general binary-input memoryless channel, which includes various important channels, such as AWGN and Laplace

channels. We show that the Bayes posterior distribution is expressed in the form of eq.(2) for turbo codes. Its extension to LDPC codes is also simple.

The information bits $\boldsymbol{x} = (x_1, \cdots, x_N)^T, x_i \in \{-1, +1\}$ and two sets of parity bits $\boldsymbol{y}_1 = (y_{11}, \cdots, y_{1L})^T$, $\boldsymbol{y}_2 = (y_{21}, \cdots, y_{2L})^T, y_{1j}, y_{2j} \in \{-1, +1\}$ are transmitted through a memoryless channel. The receiver observes their noisy version as $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2)$. Since the channel is memoryless the following relation holds

$$p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2|\boldsymbol{x}) = p(\tilde{\boldsymbol{x}}|\boldsymbol{x})p(\tilde{\boldsymbol{y}}_1|\boldsymbol{x})p(\tilde{\boldsymbol{y}}_2|\boldsymbol{x}). \tag{26}$$

The Bayes posterior with the uniform prior is

$$p(\boldsymbol{x}|\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2) = \frac{p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2|\boldsymbol{x})}{\sum_{\boldsymbol{x}} p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2|\boldsymbol{x})} = Cp(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2|\boldsymbol{x}) = Cp(\tilde{\boldsymbol{x}}|\boldsymbol{x})p(\tilde{\boldsymbol{y}}_1|\boldsymbol{x})p(\tilde{\boldsymbol{y}}_2|\boldsymbol{x}). \tag{27}$$

For memoryless channels, each conditional distribution on the right hand side of eq.(26) is formulated as

$$p(\tilde{\boldsymbol{x}}|\boldsymbol{x}) = \prod_{i=1}^{N} p(\tilde{x}_i|x_i), \quad p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x}) = \prod_{j=1}^{L} p(\tilde{y}_{rj}|y_{rj}(\boldsymbol{x})), \quad r = 1, 2. \tag{28}$$

Let us view $p(\tilde{x}_i|x_i)$ as a function of x_i, where $\tilde{x}_i$ is fixed. By defining λ_i as

$$\lambda_i = \frac{1}{2} \ln \frac{p(\tilde{x}_i|x_i = +1)}{p(\tilde{x}_i|x_i = -1)},$$

$p(\tilde{x}_i|x_i)$ is rewritten as

$$p(\tilde{x}_i|x_i) \propto \exp(\lambda_i x_i). \tag{29}$$

Note that λ_i is a function of $\tilde{x}_i$. We can also rewrite $p(\tilde{y}_{rj}|y_{rj}(\boldsymbol{x}))$ as follows.

$$p(\tilde{y}_{rj}|y_{rj}(\boldsymbol{x})) \propto \exp(\mu_{rj} y_{rj}), \quad \mu_{rj} = \frac{1}{2} \ln \frac{p(\tilde{y}_{rj}|y_{rj} = +1)}{p(\tilde{y}_{rj}|y_{rj} = -1)}, \quad r = 1, 2. \tag{30}$$

From eqs.(28), (29), and (30), eq.(27) becomes

$$p(\boldsymbol{x}|\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2) = C \exp(\boldsymbol{\lambda} \cdot \boldsymbol{x} + \boldsymbol{\mu}_1 \cdot \boldsymbol{y}_1(\boldsymbol{x}) + \boldsymbol{\mu}_2 \cdot \boldsymbol{y}_2(\boldsymbol{x})), \ \ \boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_N)^T, \ \ \boldsymbol{\mu}_r = (\mu_{r1}, \cdots, \mu_{rL})^T, \tag{31}$$

which has the identical form to eq.(2), where $c_0(\boldsymbol{x}) = \boldsymbol{\lambda} \cdot \boldsymbol{x}$, and $c_r(\boldsymbol{x}) = \boldsymbol{\mu}_r \cdot \boldsymbol{y}_r(\boldsymbol{x})$. Other distributions $p_0(\boldsymbol{x}; \boldsymbol{\theta})$ and $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)$ are also expressed with $c_0(\boldsymbol{x})$ and $c_r(\boldsymbol{x})$, which shows the information geometrical framework is valid for general binary-input memoryless channels.

Finally, we give practical form of $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}_r$ for an AWGN channel. Let the noise variance of an AWGN channel be ς^2 and $p(\tilde{\boldsymbol{x}}|\boldsymbol{x})$ becomes

$$p(\tilde{\boldsymbol{x}}|\boldsymbol{x}) = (2\pi\varsigma^2)^{-N/2} \exp\Big(-\sum_{i=1}^{N} \frac{(\tilde{x}_i - x_i)^2}{2\varsigma^2}\Big) = (2\pi\varsigma^2)^{-N/2} \exp\Big(\frac{-1}{2\varsigma^2} \sum_{i=1}^{N} (x_i^2 - 2\tilde{x}_i x_i + \tilde{x}_i^2)\Big).$$

Since $x_i^2 = 1$ holds, it becomes

$$p(\tilde{\boldsymbol{x}}|\boldsymbol{x}) = (2\pi\varsigma^2)^{-N/2} \exp\Big(\frac{1}{2\varsigma^2}(2\tilde{\boldsymbol{x}} \cdot \boldsymbol{x} - N - |\tilde{\boldsymbol{x}}|^2)\Big).$$

Following the same line for $p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})$, the Bayes posterior with the uniform prior is

$$p(\boldsymbol{x}|\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2) = C \exp(\boldsymbol{\lambda} \cdot \boldsymbol{x} + \boldsymbol{\mu}_1 \cdot \boldsymbol{y}_1(\boldsymbol{x}) + \boldsymbol{\mu}_2 \cdot \boldsymbol{y}_2(\boldsymbol{x})), \quad \boldsymbol{\lambda} = \frac{1}{\varsigma^2}\tilde{\boldsymbol{x}}, \ \ \boldsymbol{\mu}_r = \frac{1}{\varsigma^2}\tilde{\boldsymbol{y}}_r,$$

which is identical to eq.(31).

APPENDIX II
EXPLICIT FORMS OF G AND T

Metric tensor G :

for g_{ij} :

$$g_{ij} = E_{p_0}[(x_i - \eta_i)(x_j - \eta_j)] = (1 - \eta_i^2)\delta_{ij},$$

which is the diagonal matrix $I_0(\boldsymbol{\theta}^*)$.

for g_{ir} :

$$g_{ir} = \text{Cov}[x_i, c_r(\boldsymbol{x})] = \frac{1 - \eta_i^2}{\eta_i}\bar{c}_r h_{ir}, \;\; \tilde{g}_{ir} = (I_0^{-1}(\boldsymbol{\theta}^*)G_{\boldsymbol{\theta v}})_{ir} = \frac{1}{\eta_i}\bar{c}_r h_{ir}$$

Skewness tensor T :

for T_{ijk} :

$$T_{ijk} = E_{p_0}[(x_i - \eta_i)(x_j - \eta_j)(x_k - \eta_k)] = -2\eta_i(1 - \eta_i^2)\delta_{ijk},$$

where δ_{ijk} is equal to 1 when $i = j = k$ and 0 otherwise. Hence, it is diagonal.

for T_{ijr} :

$$T_{iir} = -2h_{ir}(1 - \eta_i^2)\bar{c}_r, \;\; T_{ijr} = h_{ir}h_{jr}\frac{(1 - \eta_i^2)(1 - \eta_j^2)}{\eta_i\eta_j}\bar{c}_r.$$

for T_{irs} $(r \neq s)$:

$$T_{irs} = E_{p_0}[(x_i - \eta_i)(c_r(\boldsymbol{x}) - \bar{c}_r)(c_s(\boldsymbol{x}) - \bar{c}_s)]. \tag{32}$$

When $\mathcal{L}_r \cap \mathcal{L}_s = \emptyset$, $T_{irs} = 0$. For $\mathcal{L}_r \cap \mathcal{L}_s \neq \emptyset$, we consider three cases.

case 1) $i \notin \mathcal{L}_r, \mathcal{L}_s$: In this case, x_i and $(c_r(\boldsymbol{x}), c_s(\boldsymbol{x}))$ are independent:

$$T_{irs} = 0.$$

case 2) $i \in \mathcal{L}_r$, $i \in \mathcal{L}_s$: Careful calculation of eq.(32) gives

$$T_{irs} = -2\frac{1 - \eta_i^2}{\eta_i}\bar{c}_r\bar{c}_s.$$

case 3) $i \in \mathcal{L}_r$, $i \notin \mathcal{L}_s$ or $i \notin \mathcal{L}_r$, $i \in \mathcal{L}_s$: Careful calculation gives

$$T_{irs} = \bar{c}_r\bar{c}_s\left\{-\frac{1 - \eta_i^2}{\eta_i} + \frac{1 - \eta_i^2}{\eta_i}\frac{1}{P_{rs}}\right\}.$$

APPENDIX III
EXPLICIT FORM OF $B_{rs}\eta_i$ FOR $r \neq s$

First, we give the form of $B_{rs}\eta_i$ as follows,

$$B_{rs}\eta_i = -T_{irs} - \sum_{jk} T_{ijk}\tilde{G}_{jr}\tilde{G}_{ks} + \sum_j \left(T_{ijr}\tilde{G}_{js} + T_{ijs}\tilde{G}_{jr}\right).$$

for $i \notin \mathcal{L}_r$, $i \notin \mathcal{L}_s$:

$$B_{rs}\eta_i = 0$$

for $i \in \mathcal{L}_r$, $i \in \mathcal{L}_s$:

$$T_{irs} = -2\frac{1-\eta_i^2}{\eta_i}\bar{c}_r\bar{c}_s$$

$$\sum_{jk} T_{ijk}\tilde{G}_{jr}\tilde{G}_{ks} = T_{iii}\tilde{G}_{ir}\tilde{G}_{is} = -2\frac{1-\eta_i^2}{\eta_i}\bar{c}_r\bar{c}_s$$

$$\sum_{j} T_{ijr}\tilde{G}_{js} = T_{iir}\tilde{G}_{is} + \sum_{j\neq i} T_{ijr}\tilde{G}_{js} = -2\frac{1-\eta_i^2}{\eta_i}\bar{c}_r\bar{c}_s + \sum_{j\in\mathcal{L}_r\cap\mathcal{L}_s\setminus i}\frac{(1-\eta_i^2)(1-\eta_j^2)}{\eta_i\eta_j^2}\bar{c}_r\bar{c}_s$$

Hence

$$B_{rs}\eta_i = 2\sum_{j\in\mathcal{L}_r\cap\mathcal{L}_s\setminus i}\frac{(1-\eta_i^2)(1-\eta_j^2)}{\eta_i\eta_j^2}\bar{c}_r\bar{c}_s,$$

which vanishes when $\mathcal{L}_r \cap \mathcal{L}_s$ does not include any j other than i.

for $i \in \mathcal{L}_r$, $i \notin \mathcal{L}_s$ (or $i \in \mathcal{L}_s, i \notin \mathcal{L}_r$):

$$T_{irs} = \bar{c}_r\bar{c}_s\frac{1-\eta_i^2}{\eta_i}\left(\frac{1}{P_{rs}} - 1\right),\ \ T_{ijk}\tilde{G}_{jr}\tilde{G}_{ks} = 0,\ \ T_{ijs}\tilde{G}_{jr} = 0,$$

and

$$\sum_{j} T_{ijr}\tilde{G}_{js} = \sum_{j\in\mathcal{L}_r\cap\mathcal{L}_s}\frac{(1-\eta_i^2)(1-\eta_j^2)}{\eta_i\eta_j^2}\bar{c}_r\bar{c}_s.$$

Hence,

$$B_{rs}\eta_i = \frac{1-\eta_i^2}{\eta_i}\bar{c}_r\bar{c}_s\left(-\frac{1-P_{rs}}{P_{rs}} + \sum_{j\in\mathcal{L}_r\cap\mathcal{L}_s}\frac{1-\eta_j^2}{\eta_j^2}\right).$$

When $\mathcal{L}_r \cap \mathcal{L}_s = \{j\}$, $P_{rs} = \eta_j^2$, which reduces to

$$B_{rs}\eta_i = 0.$$

ACKNOWLEDGMENT

We thank Chiranjib Bhattacharyya who gave us the opportunity to face this interesting and important issue. We are also grateful to Yoshiyuki Kabashima for his advice from the standpoint of statistical physics, and to Motohiko Isaka for his useful discussions.

REFERENCES

[1] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.

[2] T. J. Richardson, "The geometry of turbo-decoding dynamics," *IEEE Trans. Inform. Theory*, vol. 46, pp. 9–23, Jan. 2000.

[3] R. G. Gallager, "Low density parity check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

[4] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399–431, Mar. 1999.

[5] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo. CA: Morgan Kaufmann, 1988.

[6] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's "belief propagation" algorithm," *IEEE J. Select. Areas in Commun.*, vol. 16, pp. 140–152, Feb. 1998.

[7] Y. Kabashima and D. Saad, "Belief propagation vs. TAP for decoding corrupted messages," *Europhys. Lett.*, vol. 44, pp. 668–674, Dec. 1998.

[8] Y. Kabashima and D. Saad, "Statistical mechanics of error-correcting codes," *Europhys. Lett.*, vol. 45, pp. 97–103, Jan. 1999.

[9] Y. Kabashima and D. Saad, "The TAP approach to intensive and extensive connectivity systems," in *Advanced Mean Field Methods – Theory and Practice* (M. Opper and D. Saad, eds.), ch. 6, pp. 65–84, The MIT Press, 2001.

[10] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. on Inform. Theory*, vol. 47, pp. 599–618, Feb. 2001.

[11] S. Amari, *Differential-Geometrical Methods in Statistics*, vol. 28 of *Lecture Notes in Statistics*. Berlin: Springer-Verlag, 1985.

[12] S. Amari and H. Nagaoka, *Methods of Information Geometry*. AMS and Oxford University Press, 2000.

[13] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Bethe free energy, Kikuchi approximations, and belief propagation algorithms," Mitsubishi Electric Research Laboratories, Cambridge, MA, Tech. Rep. TR2001–16, May 2001.

[14] S. Amari, "Information geometry on hierarchy of probability distributions," *IEEE Trans. Inform. Theory*, vol. 47, no. 5, pp. 1701–1711, 2001.

[15] T. Tanaka, "Information geometry of mean-field approximation," *Neural Computation*, vol. 12, pp. 1951–1968, Aug. 2000.

[16] H. J. Kappen and W. J. Wiegerinck, "Mean field theory for graphical models," in *Advanced Mean Field Methods – Theory and Practice* (M. Opper and D. Saad, eds.), ch. 4, pp. 37–49, The MIT Press, 2001.

[17] S. Amari, S. Ikeda, and H. Shimokawa, "Information geometry and mean field approximation: The α-projection approach," in *Advanced Mean Field Methods – Theory and Practice* (M. Opper and D. Saad, eds.), ch. 16, pp. 241–257, The MIT Press, 2001.

[18] T. Tanaka, "Information geometry of mean-field approximation," in *Advanced Mean Field Methods – Theory and Practice* (M. Opper and D. Saad, eds.), ch. 17, pp. 259–273, The MIT Press, 2001.

[19] T. Tanaka, S. Ikeda, and S. Amari, "Information-geometrical significance of sparsity in Gallager codes," in *Advances in Neural Information Processing Systems 14* (T. G. Dietterich, S. Becker, and Z. Ghahramani, eds.), pp. 527–534, Cambridge, MA: The MIT Press, April 2002.

[20] R. G. Gallager, *Low density parity check codes*. Research Monograph series, Cambridge: The MIT Press, 1963.

[21] T. Murayama, Y. Kabashima, D. Saad, and R. Vicente, "Statistical physics of regular low-density parity-check error-correcting codes," *Physical Review E*, vol. 62, pp. 1577–1591, Aug. 2000.

[22] D. Agrawal and A. Vardy, "The turbo decoding algorithm and its phase trajectories," *IEEE Trans. Inform. Theory*, vol. 47, pp. 699–722, Feb. 2001.

[23] M. Wainwright, T. Jaakkola, and A. Willsky, "Tree-based reparameterization for approximate inference on loopy graphs," in *Advances in Neural Information Processing Systems 14* (T. G. Dietterich, S. Becker, and Z. Ghahramani, eds.), Cambridge, MA: The MIT Press, 2002.

[24] A. L. Yuille, "CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation," *Neural Computation*, vol. 14, pp. 1691–1722, Jul. 2002.

LIST OF FIGURES

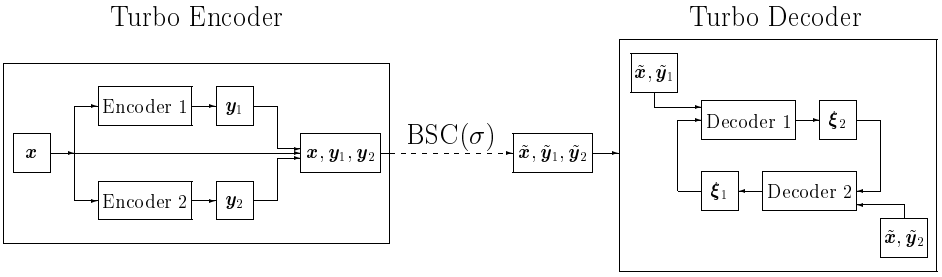

Fig. 1. Structure of turbo codes.

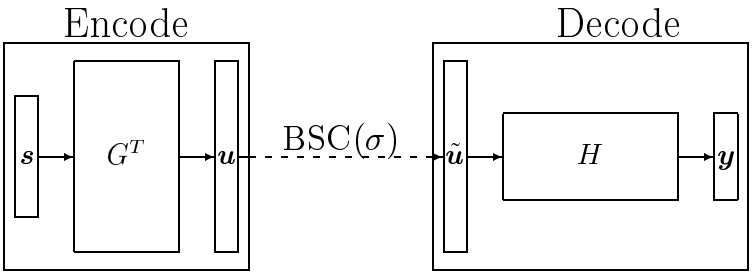

Fig. 2. Structure of LDPC codes.

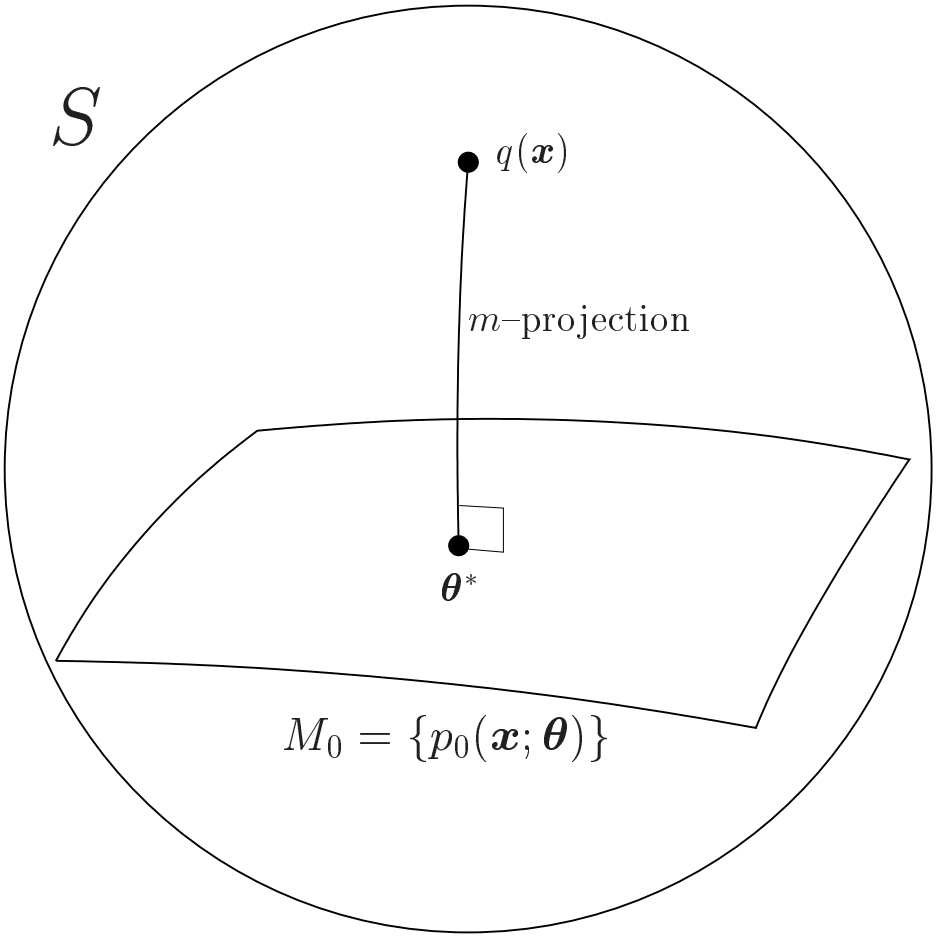

Fig. 3. Information geometry of MPM decoding

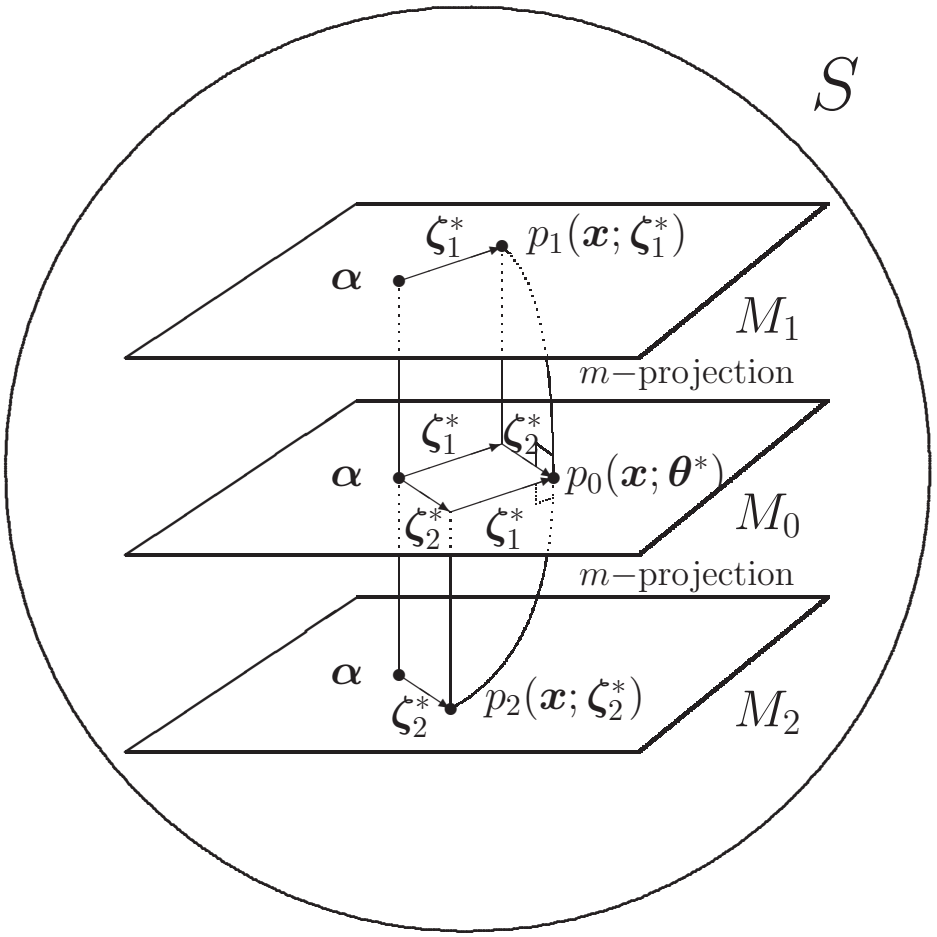

Fig. 4. Information geometrical view of turbo decoding.

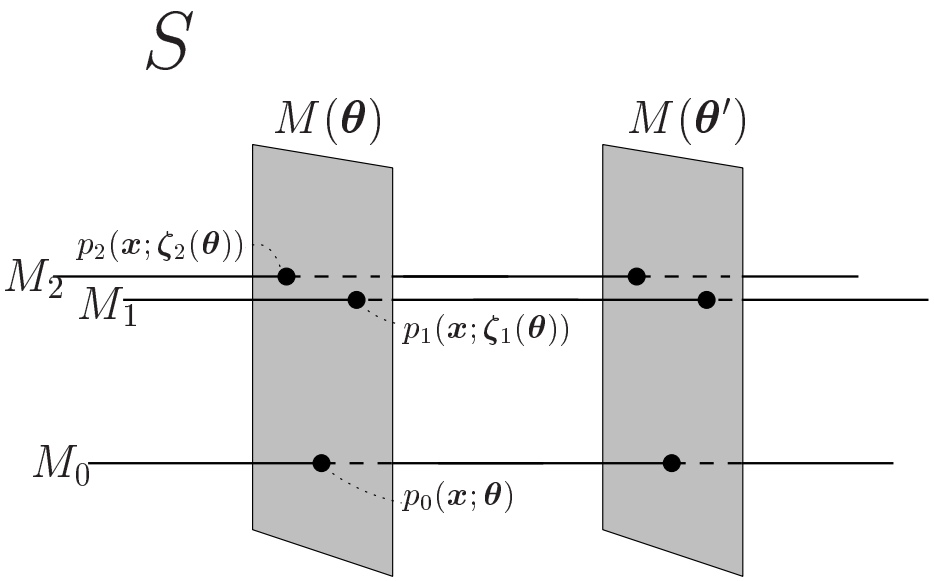

Fig. 5. Equimarginal submanifold $M(\boldsymbol{\theta})$.

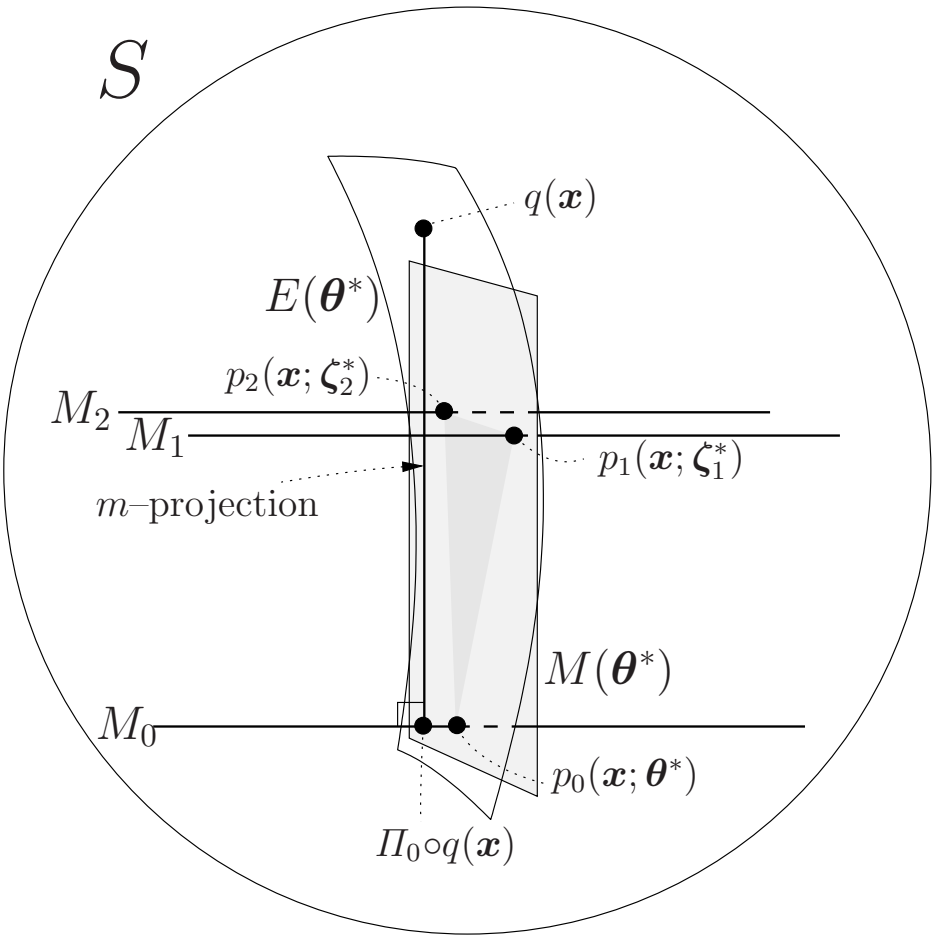

Fig. 6. $M(\boldsymbol{\theta}^*)$ and $E(\boldsymbol{\theta}^*)$ of turbo decoding: $\Pi_0 \circ q(\boldsymbol{x})$ is direct m–projection of $q(\boldsymbol{x})$ to M_0, which corresponds to true "soft decoding" based on $q(\boldsymbol{x})$, while $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$ is equilibrium of turbo decoding. Discrepancy between two submanifolds causes the decoding error.